LXF3(77) Linux Format Март 2006

BSD Linux – не единственная открытая ОС с.44

AUTOPACKAGE Первое пошаговое руководство с.84

XEN Властелин виртуальных машин с.48

LINUX FORMAT

ЧИТАЮТ ВСЕ ПОЛЬЗОВАТЕЛИ LINUX



GOOGLE НЕ ЗЛО?

Лето кода закончилось: **Крис ДиБона** объясняет, почему Google не станет открытым на 100% с.40

К в@шим услуг@м

Ваш Linux может все!

IRC ▸ Интернет-радио ▸ Jabber

BitTorrent ▸ WebDAV



№3 (77)

март 2006

ОБЗОРЫ ДЛЯ ВАС

Испытателями в Linux Format могут стать лишь избранные! Читайте в этом номере:

KDEVELOP 3.3 | **APACHE 2.2** | **AUDACITY 1.3**

X2 | **NETBSD** ПЛЮС сравнение астрономических программ и словарей

«Особенность Open Source в том... что ему не нужны спонсоры, ему нужны люди, которые создают код.» **Крис ДиБона**



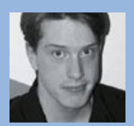
4 607130 820056

| К ВАШИМ УСЛУГАМ | ЗАПУСКАЕМ XEN | СОЗДАЕМ ПАКЕТ AUTOPACKAGE | BSD – ДРУГАЯ ОС | KDEVELOP | КРИС ДИБОНА | СРАВНЕНИЕ: АСТРОНОМИЯ

LINUX FORMAT

К ВАШИМ УСЛУГАМ...

Этот месяц редакция LXF целиком посвятила серверам, что навело нас на вопрос: «Какую деятельность, вещь или человека вы бы предпочли заменить Linux-сервером?»



Пол Хадсон
«Меня. Я не справляюсь с работой – слишком занят Mario 64»



Грэм Моррисон
«Я бы заменил неприветливых барменов пингвинистыми пив-серверами»



Эфрейн Эрнандез-Мендоза
«Португальскую, ангольскую и итальянскую сборную по футболу»



Майк Сондерс
«Судебную систему. Можно собрать Сервер Наказаний, который будет сурово карать убийц»



Ребекка Смолли
«Глаженье кошек. Я думаю, компьютер был бы куда более добросовестным, чем все владельцы животных.»



Эндрю Грегори
«Авторов сценариев последнего поколения. чушь+дурь=сюжет»



Дэвид Картрайт
«Продавцов автомобилей. Машины, ремонт и запчасти стали бы бесплатными. apt-get карбюратор»

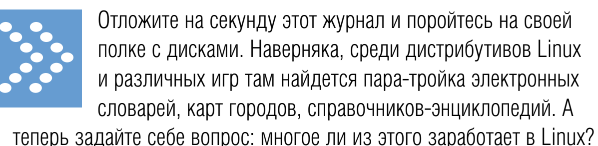


Энди Ченнел
«Было бы занятно попробовать это со спортинвентарем – боксерскими перчатками, дротиками, мячами»



Майкл Дж. Хэммел
«Планирование и приготовление ужина. Мне это уже надоело»

НЕФОРМАТНЫЙ CLOSED SOURCE



Отложите на секунду этот журнал и поройтесь на своей полке с дисками. Наверняка, среди дистрибутивов Linux и различных игр там найдется пара-тройка электронных словарей, карт городов, справочников-энциклопедий. А теперь задайте себе вопрос: многое ли из этого заработает в Linux?

Возможно, кое-что из обнаруженного и запустится в Wine, но давайте посмотрим на проблему шире. На этих дисках содержится, прежде всего, не программа, а база данных в некотором формате. Является ли он открытым? Скорее всего, нет.

Производители проприетарного ПО испытывают какой-то суеверный страх перед публикацией форматов данных. В случае с Microsoft это, в принципе, объяснимо: умея работать с DOC и XLS, можно создать продукт, напрямую конкурирующий с Office, что мы и наблюдаем на примере OpenOffice.org. Но что теряет производитель словаря от публикации его формата? Весьма вероятно, что найдутся желающие написать новую программу-оболочку, и для Linux, и даже для Windows. Но разве она будет конкурентом диску, который вы сейчас держите в руках? Нет – ведь для ее работы вам все равно потребуется CD с базой данных. Производитель же от этого только выиграет: во-первых, наличие оболочки для других платформ приведет к расширению пользовательской базы, а во-вторых, если новая программа окажется удачной, можно будет поддерживать ее официально и, таким образом, сократить затраты на разработку. Конечно, системы защиты от копирования, которыми зачастую снабжены справочники, «сидят» именно в исполняемых модулях, но вряд ли в нашей стране найдется много желающих скачивать из Интернета пиратские копии баз в объеме одного-двух CD, если их легальная стоимость – несколько сотен рублей. В противном же случае, относительный процент «пиратов из мира Linux» вряд ли будет значительным.

Незабвенный Иван Иванович из «Чебурашки» не сомневался – если он будет все-все запрещать, про него скажут, что он злой. Создатели клипартов тоже без проблем распространяют свои изображения в открытых форматах. Быть может, пришла пора пересмотреть тактику «никому-ничего-на-всякий-случай»?



С уважением,
СИНИЦЫН ВАЛЕНТИН
главный редактор LinuxFormat в России

КАК С НАМИ СВЯЗАТЬСЯ

Письма для публикации:
letters@linuxformat.ru
Подписка и предыдущие номера:
subscribe@linuxformat.ru
Техническая поддержка:
answers@linuxformat.ru
Решение проблем с дисками:
disks@linuxformat.ru
Общие вопросы: **info@linuxformat.ru**
Web-сайт: **www.linuxformat.ru**

Адрес редакции: Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54
Телефон редакции: (812) 717-00-37
Дополнительная информация на стр.120

МИССИЯ ЖУРНАЛА

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО



Содержание



Добро пожаловать в LinuxFormat – ваш гид в мире Linux

LXF3(77) МАРТ 2006



» ЧИТАЙТЕ В ЭТОМ ВЫПУСКЕ



МЫСЛИ ВСЛУХ

«Текстуры в *Celestia* безупречны – вы даже можете видеть, как свет отражается от океанов.»



«*Deepest Sender* – то самое приложение, которого пользователи Firefox ждали годами.»



12 X2: The Threat





40 Chris DiBona

44 FreeBSD

На диске

Целый DVD интересных программ
115



» DVD

BSD ПОПУРРИ – расширьте свои знания о Unix
APACHE 2.2 – самый популярный web-сервер стал проще и быстрее
СЕРВЕРА – те самые, о которых мы говорили в спецрепортаже
ИГРЫ - NetHack и Vulture's Eye

Вырежьте свою обложку для DVD на стр. **113-114**



Перед использованием диска ознакомьтесь с инструкцией на стр. 113

Главные новости

• Торвальдс сердит • *Gimp* стукнуло десять • Последний KDE перед 4.0 • Novell и Red Hat дружат с IBM • Служба Здравоохранения выбирает Novell • Отставка инициатора перехода на формат ODF в Массачусетсе • Обновление *Gstreamer*

IBM хочет дружить с Novell и Red Hat



Адам Джолланс из IBM: «Это облегчит потребителям приобретение законченных систем Linux.»

Место за большим столом и возросшие перспективы продаж для ведущих дистрибутивов Linux

IBM заявила, что увеличение спроса на «законченные» решения Linux со стороны ее клиентов навело корпорацию на мысль о придании Red Hat и Novell высшего партнерского статуса. Обе компании стали теперь частью программы IBM «Стратегический Альянс» (Strategic Alliance), присоединившись к таким элитным корпоративным партнерам IBM, как Cisco и SAP.

Этот шаг откроет для корпоративного ПО от Red Hat и Novell новые каналы сбыта, а для оконечных пользователей упростит приобретение программ и услуг, созданных для Linux, и оборудования производства IBM. Адам Джолланс (Adam Jollans), менеджер IBM по Linux-стратегии, сообщил, что решение о продвижении Red Hat и Novell было принято под влиянием трех факторов: «Во-первых, невероятный спрос на Linux среди потребителей по всему миру за последние шесть лет. Во вторых, неуклонное развитие ключевой роли Red Hat и Novell как поставщиков дистрибутивов… на фоне эры бурного роста. И, в-третьих, объем запросов со стороны покупателей, которые хотели бы приобретать Linux вместе с оборудованием и сервисами».







IBM надеется, что этот союз позволит познакомить новое поколение корпоративных покупателей с преимуществами ОС. «Это упростит для покупателей, особенно из малого и среднего бизнеса, процесс приобретения полных Linux-систем, тем самым увеличив долю его использования», – сказал Джолланс. «Это заставит независимых разработчиков ПО переносить свои программы под Linux, если они этого еще не сделали, и ссылаться на Linux как на основную платформу, если сделали».

В частности, Novell и Red Hat уже предложен доступ к каналам сбыта IBM и к так называемым «центрам инноваций» на возникающих рынках – России, Китая, Индии, Бразилии и Южной Кореи.

До заключения соглашения, независимые разработчики могли установить и сконфигурировать Linux на устройствах IBM, но теперь создается новая команда по сбыту, которая будет заниматься продажами серверных приложений Novell или Red Hat вместе с оборудованием и поддержкой IBM. «Мы дали группам сбыта Red Hat и Novell (и IBM) возможность быстро создать полный пакет всего необходимого клиенту и ускорить совершение сделок»,-заявляет Джолланс.

Джолланс также считает, что подобного рода сделки всегда открывали новые возможности. «Шесть лет назад, когда IBM впервые вложила миллиард долларов в Linux, многие спрашивали у нас: «Когда же IBM создаст собственный дистрибутив Linux?» А мы отвечали: «Это не входит в наши планы. Мы хотим стать частью Linux-сообщества и будем работать с уже существующими поставщиками дистрибутивов Linux, такими, как Red Hat и SUSE. Этот подход оказался весьма успешным.»

Новый президент Novell Рон Овсепян (Ron Hovsepian) заявил, что новый высокий статус упрочил отношения фирмы с IBM и обозначил собой «новый решительный шаг вперед» по направлению к внедрению Linux в промышленную среду.



В Великобритании у IBM один центр инноваций, но гораздо выгоднее продвигать Linux через центры при возникающих рынках сбыта.

«Linux – это мейнстрим»

Реакция широких промышленных кругов на эту новость была положительной. Тони

Лок (Tony Locke), ведущий ИТ-аналитик Bloor Research, высказал мнение, что это упрочит ведущую позицию, которую занимают Red Hat и Novell в Линуксландии, но для IBM это рядовой бизнес. «Компания давно признала, что Linux мог бы играть ключевую роль в деловом мире, и продвижение Red Hat и Novell – просто формальное проявление этой веры», сказал он. «Можно не сомневаться, что все три фирмы смогут создать солидные решения с открытым кодом, готовые для коммерческого использования».

Лок отметил, что появился новый, гибкий способ для предприятий перейти на системы, основанные на Linux, но важнее, возможно, потрясающие воображение последствия, которые решение IBM будет иметь для промышленного сектора, работающего под Linux. «Вот еще одно доказательство для рынка, что Linux представляет собой надежную промышленную платформу, а не просто интересный программный продукт. Linux – это мейнстрим, и договор демонстрирует возможности открытого ПО донести свои ценности до потребителя из реального мира».

Вершина дерева

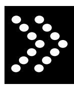
Иэн Брэмли (Ian Bramley), ведущий аналитик Software Strategies, согласен с тем, что сегодня Linux является важным игроком на рынке серверов, даже, пожалуй, предпочтительной платформой для ISV, и поддержка IBM лишь доказывает это.

«Главными партнерами IBM по ПО становятся лишь те, чей бизнес наиболее стратегически значим для этой фирмы»,- считает Брэмли. Возвышение Red Hat и Novell подчеркивает, насколько важным для IBM стал Linux. И, конечно же, IBM сегодня – самый важный канал сбыта для Red Hat и Novell, данный шаг еще раз это подчеркнул».

Со своей стороны, Novell и Red Hat будут продвигать промежуточное ПО от IBM, включая пакет web-сервисов на основе *Geronimo* J2EE от Apache Foundation и базу данных *Derby*, которые будут присутствовать в следующей итерации промышленного сервера SUSE Linux – и в большей степени будут вовлечены в разработку и продвижение платформы *Eclipse*. Red Hat также сертифицирует сервер IBM Websphere Community Edition для Red Hat Enterprise Linux.

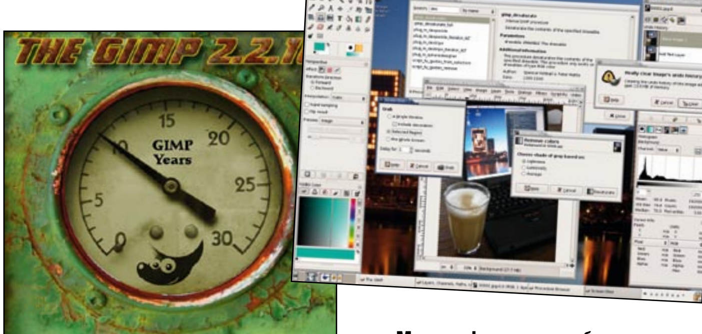
С днем рождения, Wilbur!

Редактор изображений Gimp празднует десятилетие



***Gimp*, стандартное приложение** Linux для редактирования изображений, вступил во второе десятилетие активной деятельности с новым юбилейным релизом и красочным экраном, созданным победителем в конкурсе 'ix'.

Питер Маттис (Peter Mattis), который начинал этот проект вместе со Спенсером Кимболлом (Spencer Kimball), объявил о выходе приложения в 1995 году. Его разновидности имелись для Linux 1.2.13, Solaris 2.4, HPUX 9.05 и SGI IRIX. Первая бета-версия под номером 0.54 была выпущена в феврале 1996-го, она использовала библиотеку Motif, но работа с коммерческим набором инструментов не приводила в восторг открытое сообщество, и программисты создали собственный проект *Gimp Tool Kit* (GTK). К концу 1996 года была выпущена полная версия *Gimp 0.99* с поддержкой слоев и прочной системой управления памятью.



Красочный юбилейный экран, отмеченный талисманом Wilbur.

Многие фильтры снабжены пред-просмотром с масштабированием.

Версия 1.0 – первая не-бета-версия *Gimp* – увидела свет 19 мая 1998 года. Популярность приложения в сообществе FOSS обеспечило быстрое расширение библиотеки подключаемых модулей.

В дополнение к юбилейному выпуску появился и снимок новой версии, 2.3.6, чей выход совпал со временем проведения 22-го конгресса Chaos Communication Congress в Центре конференций в Берлине, Германия. Последняя версия уменьшила время запуска и позволяет использовать колесо прокрутки мыши для просмотра, увеличения и уменьшения размера изображения в фильтрах пред-просмотра; улучшился также доступ с клавиатуры к инструментам типа размер и форма кисти. Браузер справки тоже изменен во имя удобства пользования.

LVEE 2006 НА ПОДХОДЕ – СЛЕДИТЕ ЗА ИНФОРМАЦИЕЙ НА http://lvee.open.by

Minsk Linux Users Group приглашает посетить мероприятие, которое пройдет 25-28 мая под Минском в пансионате «Звездный» и называется Linux Vacation East Europe (LVEE) 2006.

Что же прячется за таким названием? Понятно, что это отдых для единомышленников, занятых в области свободного программного обеспечения, и в частности Linux, либо как-то связанных с ней. Мы предлагаем Вам провести 4 незабываемых дня среди единомышленников. В прошлом году мероприятие прошло под Гродно и привлекло около 80 разработчиков/внедренцев из России, Украины, Беларуси и Польши.

Предварительный приоритет при регистрации имеют следующие категории граждан:

- Участники и контрибуторы Free Software/OpenSource проектов
- Люди, подготовившие тезисы и доклады к конференции
- Специалисты, не подготовившие доклады, но способные своим участием в дискуссиях повысить уровень конференции
- Участники подготовки фестиваля

Следите за информацией на сайте!

Дэвид Картрайт



Дэвид Картрайт (David Cartwright) – ИТ-консультант, специалист по системам и решениям Linux.

КОММЕНТАРИЙ

Лучше не спорьте

И снова пришлось мне возиться с межплатформными проблемами, потому что кто-то решил поумничать. Один из моих клиентов купил систему записи звонков для своего call-центра: здоровый сексуальный (ну ладно, серый) сервер с огромным дисковым пространством и крутым программным обеспечением, которое превращает входящие звонки в WAV-файлы на диске. Очень милый графический интерфейс пользователя со всевозможными поисками по звонившему, по расширению, по времени звонка – чего там только нет, а поскольку все основано на web, можете загрузить его и прослушать запись с любого компьютера.

Беда в том, что вы этого не можете. Вместо того, чтобы соединиться с WAV-файлом, система загружает занудный компонент ActiveX, и он озвучивает файл – или не озвучивает, если на вашем компьютере нет Windows.

В таком случае, да здравствует FreeTDS (**www.freetds.org**), открытый клиент-серверный протокол *SQL Server* ! Поскольку в роли сервера у моего клиента выступает ПК с Windows, понятно, что до хранящихся на нем файлов можно добраться с помощью *Samba*. Все, что надо сделать – это соединиться с базой данных *SQL Server* (где хранится информация о звонках для вышеупомянутого устройства поиска), скажем, с обычной PHP-страницы, и мы сможем напрямую соединить наших пользователей, у которых нет Windows, с WAV-файлами, откуда их браузеры запустят подходящий проигрыватель.

Я не утверждаю, что FreeTDS – самая легкая в мире вещь для установки, но уж если вы ее установили, она работает как часы. Открытое ПО снова приходит на помощь, когда некоторые разработчики решений для определенных платформ норовят умничать и как бы случайно создают свои продукты только для Windows.



Майкл Микс
Майкл (Michael Meeks) – штатный сотрудник Novell по разработке OpenOffice.org, он не любит селедку.

КОММЕНТАРИЙ

Репортаж из лагеря загрузки

Последние несколько месяцев я трудился над улучшением довольно вялой «горячей» загрузки **OpenOffice.org**. Она отнимает добрых пять секунд, а это чудовищно медленно. Тесты показали, что более 50% этого времени уходит на компоновку, при этом CPU почти не работает, кэши L1 и L2 простаивают. C++ обостряет эту проблему, генерируя огромное число релокаций для таблиц виртуальных функций.

Компоновка в Linux изуродована «вставками»: символам разрешено замещать друг друга в библиотеках; выходит, что для разрешения любого символа надо провести поиск во всех библиотеках (а их в *Ooo* штук 150). Добавив новую функцию прямой компоновки, можно уменьшить время разрешения ссылок на 75% или около того, почти вполовину сократив время горячего запуска *Ooo* за счет потери «вставок». В любом случае, последующий анализ показал, что вставки по большей части делаются непреднамеренно и они довольно опасны.

Усовершенствование компоновки нарушает границы *prelink*, являющегося текущим малоприятным решением проблемы. Для эффективной работы надо заново запускать *prelink* для каждого исполняемого файла, когда бы ни обновлялась библиотека. Это способствует фрагментации диска, пагубной для производительности, и вдобавок неэффективно для динамических библиотек, на которые опирается большая часть кода *OOo*.

После публикации результатов к Novell обратился независимый разработчик ПО, который пытался заставить свое большое приложение на C++ работать и под Linux, и под Win32, и столкнулся с той же проблемой. Будем надеяться, что группу поддержки *Glibc* убедят принять эту оптимизацию.



Novell: есть контракт с NHS

Крупная трехгодичная сделка на сумму около € 21 000 000 для SUSE

Novell заключила крупный контракт на поставку программного обеспечения и услуг Национальной Службе Здравоохранения Великобритании (National Health Service – NHS). Обсуждается новая трехлетняя сделка по переводу большей части информационной инфраструктуры NHS со старой системы Novell Netware на SUSE Enterprise Server, и хотя пока еще не было планов по переходу на рабочие столы Linux, Novell верит, что этот переход вполне логичен в будущем.

Сделка является частью инициативы NHS «Связи для Здравоохранения» (Connecting for Health), и Novell призван обеспечивать безопасность, управление рабочим столом и программы идентификации на 600 000 рабочих станциях организации. Дэйв Николс (Dave Nicholls), менеджер NHS в Novell, сообщает, что эта сделка – продление предыдущего трехгодичного договора, но она куда более обширна, поскольку в нее входит решение наиболее важных для здравоохранения задач. «Этот договор предоставит всей службе доступ к нашим продуктам по идентификации и управлению ресурсами и [предлагает] в будущем гибкую возможность перехода на открытое ПО и Linux».

Этот договор также означает большую экономию для NHS, и, как надеется Николс, станет хорошей возможностью для широкомасштабной демонстрации сильных сторон программ с открытым кодом перед потенциальными клиентами. NHS – крупнейший работодатель в Европе и один из крупнейших мировых потребителей услуг в сфере ИТ.

Ричард Грейнджер (Richard Granger), генеральный директор NHS по ИТ, заявил, что в основе решения остаться с Novell лежал принцип «value for money», и что предлагаемая система поможет снять барьеры на пути перехода на Linux в организации в будущем. «Этот договор обеспечивает доступ к открытым платформам промышленного класса, и вместе с этим – что еще более важно – поддержку и обучение персонала NHS для перехода на ПО с открытым кодом по разумной цене».

Novell считает, что организациям, подобным NHS, постоянно требуются программы с открытым кодом, одно из преимуществ которых –организация не обязана дожидаться обновления программы ее разработчиком, производя обновления самостоятельно со скоростью, которая устраивает организацию. Том Франчезе (Tom Francese), президент Европейского отделения Novell, полагает, что этот договор позволит NHS «получить отдачу с каждого вложенного фунта, и в то же время упростит для тысяч докторов, медсестер и другого персонала NHS задачу улучшения обслуживания населения Великобритании».



Том Франчезе (Tom Francese) из Novell считает, что общественный сектор жаждет стандартных решений с открытым кодом.

> «ОРГАНИЗАЦИЯМ ТИПА NHS ТРЕБУЮТСЯ ПРОГРАММЫ С ОТКРЫТЫМ КОДОМ.»

НОВОСТИ ВСТРАИВАЕМОГО LINUX

• Согласно данным Venture Data Corp (VDC), встроенный Linux стал 100-миллионной индустрией в 2005 году, и в 2006 году ожидается дальнейший рост. Решение таких фирм, как WindRiver и Texas Instruments, полностью перейти на открытый код стало важным фактором для принятия ОС в солидных кругах, считают аналитики VDC.

• MontaVista, один из лидеров развития встроенного Linux, объявил о своих планах добавить в свой основной продукт свойства реального времени. WindRiver и LynuxWorks уже предлагают решения Linux в реальном времени (с использованием различных методов), но MontaVista является одной из крупнейших фирм, поддерживающих встроенный Linux, и она будет продвигать включение своих функций реального времени в основное дерево ядра Linux.

• Вышли в свет две новых видеосистемы на основе Linux, и это еще более упрочнит позиции ОС. Корейское отделение Neugent Technologies выпустило новые серии своего Hawkeye DVR и объявило о намерениях создать малобюджетный DVR на встроенном Linux, а Video Without Boundaries (VWB), Флорида, создал амбициозный медиацентр на основе Linux под названием MediaReady, выполненный с поддержкой *iTunes*, iPod и Google Video Search. «Теперь у владельцев MediaReady будет доступ к наиболее популярным приложениям и сервисам, доступным онлайн прямо из своих гостиных, с удобного дивана», сообщил директор VWB Джеффри Харрелл (Jeffrey Harrell).





Ричард Коббетт **(Richard Cobbett)** Создатель флорентийского телескопа, пионер в физике, воитель церкви. Или это был Галилей?

КОММЕНТАРИЙ

Снова на арене

Итак, очередной раунд историй о GoogleOS пришел и прошел – оказалось, что это просто сны наяву мудрецов от технологии. Не то что я удивлен или там разочарован. Поймите меня правильно: я вовсе не утверждаю, что их мечта никогда не сбудется (разве что ее объявят противозаконной за день до прочтения вами этих строк), но реальных оснований для этого в природе не существует.

Любой крупный успех Google основывался либо на поиске, либо на рекламе, а любое предложение по загрузке – Google Earth, Picasa, даже новый Google Pack – созданы под Windows. В итоге нам остается… а собственно, что? Копия *Firefox* с несколькими встроенными закладками. Фактически, Knoppix. Может, Google выпустить настроенную версию Knoppix? И вам удастся сохранить целых пять нажатий на клавиши при загрузке *Gmail*?! Удержать бы первую страницу!

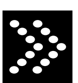
По мне, так ешьте сами этот GoogleOS. С радостью пожертвую 0,68 доллара, заработанных благодаря AdSense, на бейсбольную биту побольше, чтобы отделать всех предсказателей, забегающих впереди паровоза. Хоть бы он и вышел, посмотрите на Google Talk. Посмотрите на Orkut. Google и то не всегда попадает в десятку.

До тех пор, пока не будет официального уведомления, GoogleOS – просто мечта фановой трубы, причем эта труба плачет по бутылке ядреного растворителя, чтобы прочистить гадость, осевшую в ней за прошлый год. Или даже по толовой шашке.

И вообще, есть же Google Calendar, именно он должен потрясти мир.

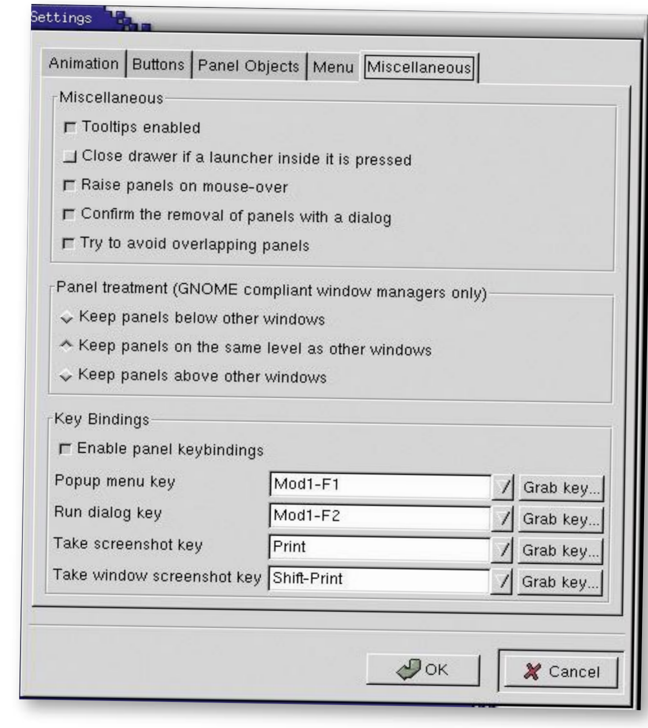
Реплика от Торвальдса: «Просто используйте KDE!»

Слово дипломату Линусу



Своим заявлением, заранее гарантирующим ссору разработчиков Gnome с пользователями и вспышку былых войн за рабочий стол, Линус Торвальдс встрял в дискуссию Gnome и KDE. При обсуждении вопросов удобства пользователя (ни больше ни меньше, как в рассылках Gnome), Торвальдс сообщил, что он собирается всего-навсего «поощрять людей выбирать KDE», добавив для полноты картины, что разработчики Gnome держат пользователей за идиотов.

«Типичная ментальность Gnome – «пользователи – идиоты, они путаются в функциях» – это чума. Если вы считаете своих пользователей идиотами, одни идиоты к вам и придут. Я не пользуюсь Gnome, потому что, стремясь к простоте, он давным-давно элементарно перестал делать то, что мне надо». В заключение Линус объявил: «Пожалуйста, просто говорите людям: возьмите KDE».



Не все UI-элементы Gnome просты, Линус. Иных и испугаться можно!

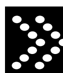
Джефф Во (Jeff Waugh), ведущий разработчик основанного на Gnome дистрибутива Ubuntu, отметил, что Gnome был создан для тех, кому не особенно интересно, как работает компьютер, но зато интересно, что можно сделать с его помощью. Однако в ответ Торвальдс предположил, что скандирование этой мантры призвано для маскировки некомпетентности или лени. «Кажется, что Gnome разрабатывали нацисты от интерфейсов, выставляющие как предлог для ничегонеделания не «это слишком сложно», а «это может запутать пользователя»».

Нэт Фридман (Nat Friedman) из Novell откликнулся на это, написав: «Нам нужно, чтобы Linux взрослел, если мы хотим его успеха на рабочих столах. Давайте обсуждать все, как взрослые люди. Будь я из Microsoft , я бы ликовал, глядя, как вы ругаетесь друг с другом». Позднее Торвальдс пожаловался, что разработчики, которые лезут из кожи вон, стараясь создать диалоги попроще – в какой-то мере F.I. (второе слово в аббревиатуре – «идиоты»).

Это послание вызвало град ответов как со стороны пользователей Gnome, так и со стороны пользователей KDE, включая дискуссию на Slashdot, которая растянулась аж на 1500 сообщений.

Форматные войны: Империя наносит ответный удар

Отставка директора по ИТ из-за лживых кляуз о превышении дорожных расходов



Питер Куинн (Peter Quinn), человек, который вынянчил идею о принятии формата OpenDocument в штате Массачусетс, ушел в отставку после того, как газета *Boston Globe* обвинила его в выездах за пределы штата на «конференции, частично спонсируемые фирмами, которые получат некие преимущества от изменений в программном обеспечении компьютеров, принятом в штате».

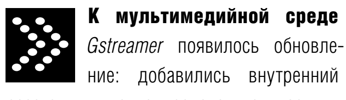
Несмотря на то, что официальное расследование его оправдало, Куинн решил, что лучше покинуть пост директора отдела информационных технологий штата, чем остаться на этом посту и стать мишенью критики за перевод делопроизводства штата с формата корпорации Microsoft *MS Office* на альтернативный ODF.

Объявив о своем решении в электронном письме персоналу, Куинн написал, что его присутствие может неблагоприятно сказаться на работе всего подразделения: «Даже самые незначительные инициативы тормозятся или полностью пресекаются теми сторонами, от которых этого совершенно не приходится ждать или которые вообще не обладают информацией». Штат Массачусетс принял решение в конце 2005 года принять в делопроизводстве файлы OpenDocument, лежащие в основе *OpenOffice.org*. Это решение было принято как для того, чтобы снять монополию поставщиков ПО, так и для того, чтобы не лишиться доступа к унаследованным документам после обновления. Однако смена форматов обрадовала не всех; в особенности, фирму Microsoft, которая позднее представила свои собственные форматы файлов в ECMA для ратификации в качестве открытого стандарта [*см. Новости, LXF76*].

Преемник Куинна, Томас Тримарко (Thomas Trimarco), заявил, что политика перехода на формат ODF будет продолжена, несмотря на уход ее инициатора с должности.

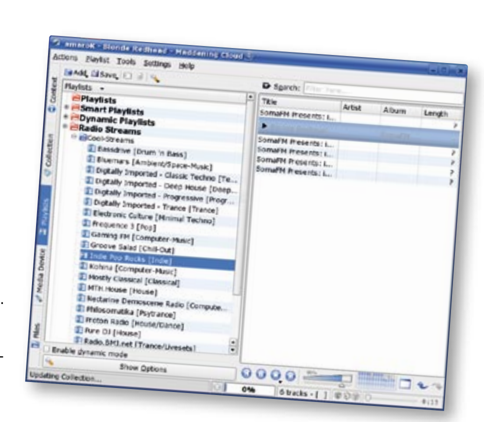
Легко сказать

Обновление мультимедиа-программ включает VoIP



К мультимедийной среде *Gstreamer* появилось обновление: добавились внутренний реестр, улучшение производительности и поддержка для Voice over IP. Поддержка VoIP появилась в результате совместных усилий (проекта Farsight, Fluendo, Nokia и бразильского Instituto Nokia de Tecnologia) по реализации протокола RTP, который, по словам разработчиков, упростит создание приложений VoIP.

Gstreamer стоит за кулисами многих приложений рабочего стола Linux, включая музыкальные проигрыватели и программы для редактирования видео. Майкл Доминик (Michael Dominic), ведущий разработчик редактора видео Diva, говорит, что *GStreamer* наконец-то стал полноценной мультимедийной программой. «Этот релиз основан на новых принципах, призванных помочь приложениям для сложной нелинейной обработки, работающим одновременно с несколькими источниками», сообщил он. «Следуя правилам хорошего дизайна, архитектура *Gst 0.10* не углубилась, а расширилась, позволяя разработчику контролировать больше аспектов реального потока данных»

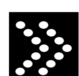


Приложения типа Amarok выиграют от обновления GStreamer.

Команды разработчиков приложений, включая мультимедийные проигрыватели *Bleep* и *Amarok*, работают над реализацией среды в новых версиях.

Планируем будущее рабочих станций

Развитие сотрудничества – ключ к успеху



Рабочая группа OSDL Desktop Linux провела свое первое крупное собрание в штаб-квартире организации в Бивертоне, штат Орегон. Разработчики и распространители ПО, в лице 70 представителей, выставили 30 проектов. Программисты от Gnome, KDE, Ubuntu, Mozilla, Free Standards Group (FSG) и прочих объединили усилия, чтобы проложить дорогу в будущее и обсудить сильные и слабые стороны ныне существующей рабочей станции Linux. Во встрече также участвовали представители Novell, Adobe, Intel, AMD, Red Hat, IBM и Nokia.

Брюс Харрингтон (Bryce Harrington) из OSDL заметил, что с середины первого же дня мероприятие зажило собственной жизнью. «С утра процесс достаточно жестко контролировали ребята из DTL [Desktop Linux], но по ходу встречи ее все больше и больше направляли и вели сами участники».

Харрингтон также подивился мирному сосуществованию «конкурирующих» проектов, таких, как KDE и Gnome и различные дистрибутивы. По его словам, они пришли к пониманию того, что вместе они сильнее, чем поодиночке. «Представители разных дистрибутивов особо подчеркнули, что благодаря сотрудничеству в таких вопросах, как совместные базы данных поддерживаемого оборудования, они общими силами добьются документации от поставщиков драйверов и прочих, и тогда Linux в целом сможет занять до 98% рынка».



Мартин Конолд (Martin Konold) на встрече представлял KDE.

Самым значимым итогом стало завершение проекта Portland, нацеленного на выработку общего технического критерия для рабочей станции Linux, учитывающего такие элементы, как драйверы оборудования и пользовательский интерфейс. Portland будет включен в Linux Standard Base (LSB) в версии 4.0, выход которой ожидается в этом году. Международная Организация Стандартизации (ISO) ратифицировала LSB в ноябре 2005 года – это шаг, который, по словам FSG, упрочит позиции OSS и у независимых поставщиков ПО, и у корпоративных покупателей.

КОРОТКОЙ СТРОКОЙ

- **Dell** начал установку ***Firefox*** на все рабочие станции и ноутбуки, которые продаются онлайн через его портал в Великобритании.



- Вызвавшее споры ПО ***David***, пообещавшее пользователям Linux работу с приложениями Windows, выставлено на продажу в Японии. *David* постоянно мелькал в заголовках в 2005 году, когда ранняя бета-версия программы вышла поразительно похожей на проект с открытым кодом *Wine*, хотя его создатели, ***SpecOps***, связь отрицают.

- Немецкий сайт *Heise Online* сообщает, что власти города **Берлина** готовы последовать примеру Мюнхена и перевести свою инфраструктуру ИТ на Linux. По плану переход начнется с серверов, а уж потом обновление коснется 58 000 рабочих станций города, ныне работающих под Windows.

- В своем ежегодном отчете Комиссии по ценным бумага и биржам, **Apple** впервые упомянула о том, что компания столкнулась с давлением со стороны рабочих станций Linux. Годовая оценка компании 10K говорит, что «Конкуренты, продающие персональные компьютеры на базе Windows и Linux, агрессивно снизили цены и опустили планку для персональных компьютеров».

- **Гвидо ван Россум** (Guido van Rossum), создатель **Python**, перешел на работу в Google.

- Вышла в свет финальная версия ***Seamonkey***, проекта, который не дает угаснуть мечте о пакете Mozilla. Говорят, что его можно использовать в Windows, но варианту для Linux уделено меньшее внимание.

КОРОТКОЙ СТРОКОЙ

• Команды разработки ***MS Internet Explorer, Firefox*** и ***Konqueror*** работают над стандартизацией элементов пользовательского интерфейса, предоставляющих информацию о безопасности, мошенничестве и RSS. *IE7* возьмет иконку RSS от *Firefox*.

• Спустя два года от начала разработки **IceWarp Software** вышла Linux-версия пакета коммуникаций уровня предприятия. ***Merak Email Server for Linux*** предоставляет обработку всех стандартных почтовых протоколов, а также защиту от спама, службу IM и встроенный FTP и web-серверы. Пакет сертифицирован для использования с RHEL.

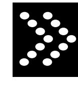
• **New Breed Software** обновила ***TuxPaint***, отличную программу для детей. Теперь в нее добавлены новые свойства, эффекты и штампы, и приложение стало доступно для трех основных платформ.



• Обновлен **WordPress**, сервер сетевых дневников. Полностью перестроены внешний и внутренний интерфейсы. Пользовательский интерфейс следует традициям приложений типа Flickr и Delicious.

В Блендере

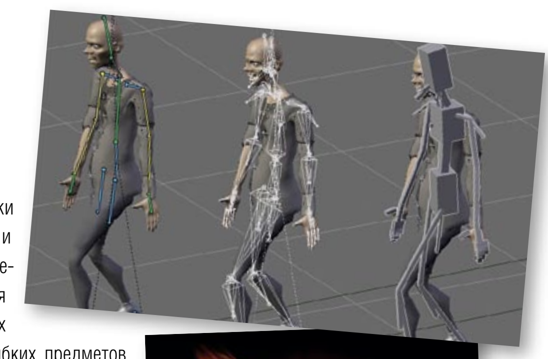
Разработчики *Blender 2.40* оценивают изменения



Один из разработчиков Blender, программы-редактора трехмерной графики, рассказал *LXF* о том, как решение об открытии его исходного кода в 2002 году повлияло на проект. Том Масгроув (Tom Musgrove), член команды MakeHuman, занимающейся усовершенствованием анимации персонажей, сообщил, что открытый код изменил и скорость разработки, и процесс управления.

«В приложениях с закрытым кодом чувствуется очень сильное давление, направляющее процесс разработки туда, куда диктует рынок», сказал он. «Когда работаешь с открытым кодом, то просто нет отдела маркетинга, который приказывает тебе, что надо делать; важность того или иного шага определяется непосредственным общением с художниками и другими пользователями, а также заплатками и предложениями разработчиков».

Blender 2.40, который рассматривался в *LXF76*, обладает рядом новшеств, делающих приложение более пригодным для анимации персонажей, включая улучшенную поддержку инверсной кинематики и усовершенствование телодвижений персонажа. Эта версия также имеет новую систему потоковой динамики для рендеринга воды и других жидкостей и систему волосяных частиц для создания реалистичных волос, травы и других гибких предметов. Добавления, хорошо принятые 3D-аниматорами, приближают *Blender* к его коммерческим конкурентам вроде *Maya*.





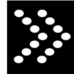
(Вверху) **Подвижность персонажа тщательно продумана и анимация стала проще** (внизу).

Разница с проприетарными временами, сказал Масгроув, заключается в том, что если пользователю нравится некое свойство, он может сам его запрограммировать. Один из таких примеров в 2.40 – заплатка, позволяющая создавать анимированные волосы и мех, которую прислал Янне Карху (Janne Karhu). Другое преимущество – возможность брать функции из других приложений и добавлять их в проект; пример – система моделирования динамики потока жидкости Нильса Тюрея (Nils Thuerey), начавшаяся как исследовательский проект, а потом добавленная в *Blender* благодаря финансовой поддержке Google в рамках инициативы Лето за Кодом (Summer of Code).

Однако развитие программ с открытым кодом отнюдь не обходится без проблем, считает Масгроув. «Если нужно выполнить какую-то конкретную разработку или административную работу, нет никакой гарантии, что найдется тот, кто захочет – и сможет – взять на себя решение задачи за тот срок, который вы отводите на ее решение».

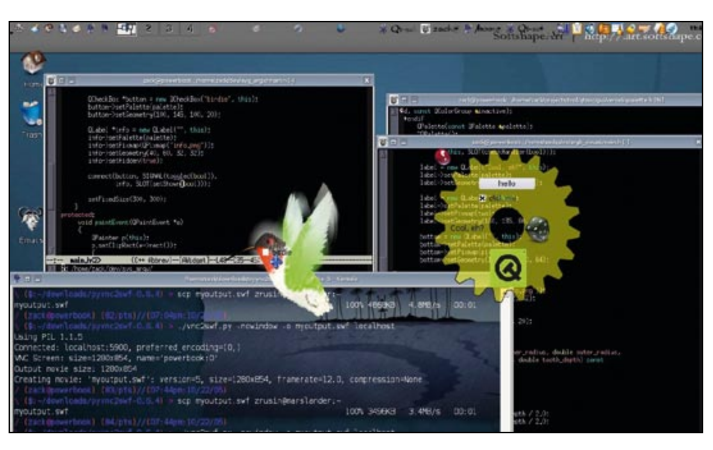
Все это о слоях

KDE 4 понял намек редакторов изображений



Вырисовываются планы фундаментальных изменений в способ отображения элементов рабочего стола в KDE 4. Новая версия, выход которой ожидается к концу года, должна также быть совместима с объектами Apple OS X *Dashboard*.

Зак Рузин (Zack Rusin), ведущий разработчик KDE в Trolltech, сообщил, что он сформулировал способ добавить слой совместимости с *Dashboard* в KDE 4, и это стало первым шагом в реформе компоновки рабочего стола. «Наш рабочий стол будет состоять из слоев, каждый из которых может быть создан подключаемым модулем», – написал Рузин в своем дневнике.



KDE 4 будет поддерживать окна неправильной формы, вроде шестерни или птички.

Идея напоминает процесс создания изображений в *Gimp*. Нижний слой – фон рабочего стола, следующими слоями будут иконки, объекты рабочего стола, эффекты слоев и так далее. Рузин сказал, что система будет «ограничена только количеством перестановок слоев, укладывающимся в голове. Никаких трюков – все официально».

Объекты Apple Dashboard – это небольшие апплеты, созданные с использованием JavaScript и HTML, а поскольку и KDE, и *Safari* (web-браузер операционной системы X) созданы с использованием движка рендеринга *KHTML*, межплатформная поддержка для них особой проблемы не составит. Web-сайт Apple содержит сотни виджетов, которые занимаются всем – от показа Google Maps до игр и поисковых утилит, но не все они будут работать в KDE, поскольку некоторые используют родной для OS X движок *Cocoa*. Однако разработчики могут переделать несовместимые виджеты с использованием родных опций KDE. LXF

Linux в желто-красной гамме

Всем известно, что одним из наиболее серьезных факторов, сдерживающих проникновение Linux на настольные компьютеры является отсутствие поддержки этой операционной системой ведущими производителями делового софта. В нашей стране самым крупным разработчиком программного обеспечения для бизнеса является фирма «1С», продуктами которой давно и успешно пользуется многомиллионная армия управляющих менеджеров, бухгалтеров, кадровиков и т.д. До недавнего времени они даже не рассматривали переход на Linux по причине отсутствия соответствующих программных средств от «1С». В скором времени ситуация может измениться в положительную сторону.

В конце февраля состоялся очередной партнерский семинар «1С». Подобные мероприятия проводятся регулярно и являются эффективной формой взаимодействия руководства с партнерами-франчайзи. На семинарах партнеры узнают о результатах работы «1С» за прошедший год, знакомятся с новыми продуктами и с ближайшими планами.

Одной из ключевых тем семинара стали обнародованные «1С» рабочие планы по развитию технологической платформы «1С:Предприятие». К концу июня текущего года должна увидеть свет очередная бета-версия платформы под номером 8.1. По сравнению с существующей версией, она претерпит архитектурные и технологические изменения, ориентированные на расширение функциональности, оптимизацию работы, обеспечение большей масштабируемости (в том числе, поддержку кластера серверов), расширенную поддержку XML, полноценную реализацию веб-сервисов. И, что наиболее актуально для всех приверженцев свободной операционной системы, в этой версии впервые будет реализована работа сервера «1С: Предприятия» под Linux с поддержкой СУБД PostgreSQL. Напомним, что Windows-версия базируется на СУБД MS SQL Server. Пока не объявлены планируемые цены на сервер «1С: Предприятия 8.1» для Linux, этот вопрос находится в стадии рассмотрения, в остальном же ценовая политика измениться не должна.

Это сообщение вызвало бурный интерес к Linux-решениям у участников семинара. Собственно, они уже могли ознакомиться с работой текущей версии «1С:Предприятия» в среде Linux. Так, на семинаре была представлена разработка компании Etersoft, представляющая собой специальную сборку среды Wine, предназначенной для запуска в Linux различных Windows-приложений. В частности, она позволяет запускать сетевые версии «1С: Предприятия». Однопользовательские версии, к сожалению, пока запускать невозможно по причине отсутствия поддержки аппаратных HASP-ключей. www.mysqluc.com

Григорий Рудницкий

ASPLinux 11 Deluxe от LinuxCenter

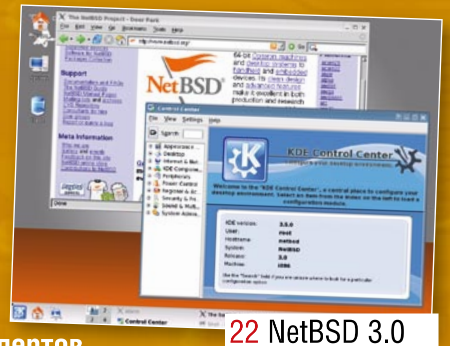
ASPLinux 11 Deluxe – универсальный и самый полный вариант дистрибутива ASPLinux 11. Разработчики и системные администраторы с успехом могут использовать дистрибутив, как для на рабочей станции, так и в качестве сервера небольшой сети.

Благодаря наличию полного комплекта документации, исходных кодов и продолжительной технической поддержке дистрибутив идеально подойдет для глубокого изучения Linux с любого уровня – как новичками, так и желающими освоить навыки администрирования и разработки. В состав ASPLinux 11 Deluxe входит три печатных руководства – Администратора, Пользователя, Руководство по установке.

В комплект также включены игры и демо-версии коммерческих приложений под Linux. Все необходимые приложения и модули, кодеки и обновления пакетов уже находятся на установочных дисках и после инсталляции пользователь получает полноценную рабочую среду сразу, независимо от того, как компьютер будет использоваться – для работы в Интернет, воспроизведения медиафайлов, создания документов, как серверная система или любым другим образом.



Обзоры >>


22 NetBSD 3.0

Новинки программного и аппаратного обеспечения с точки зрения наших экспертов

ПОЛЬЗОВАТЕЛЯМ LINUX НЕ ХВАТАЕТ PHOTOSHOP?


Евгений Балдин
Научил Linux говорить по-русски.

Ещё не хватает 1С, Autocad и программы класса FineReader, только Open Source, пожалуйста. Кроме перечисленного не нравится «техническая поддержка Linux – посылают все тебя с твоим вопросом на три буквы (man)». Интересно, а куда посылают в случае не Linux? Здесь хоть место приличное.

«Бедные пользователи Linux», – может подумать сторонний наблюдатель, случайно изучивший две оживлённые дискуссии на (LOR) http://www.linux.org.ru/, посвящённые опросу компании Novell: «Какое программное обеспечение вы хотите видеть в Linux? »

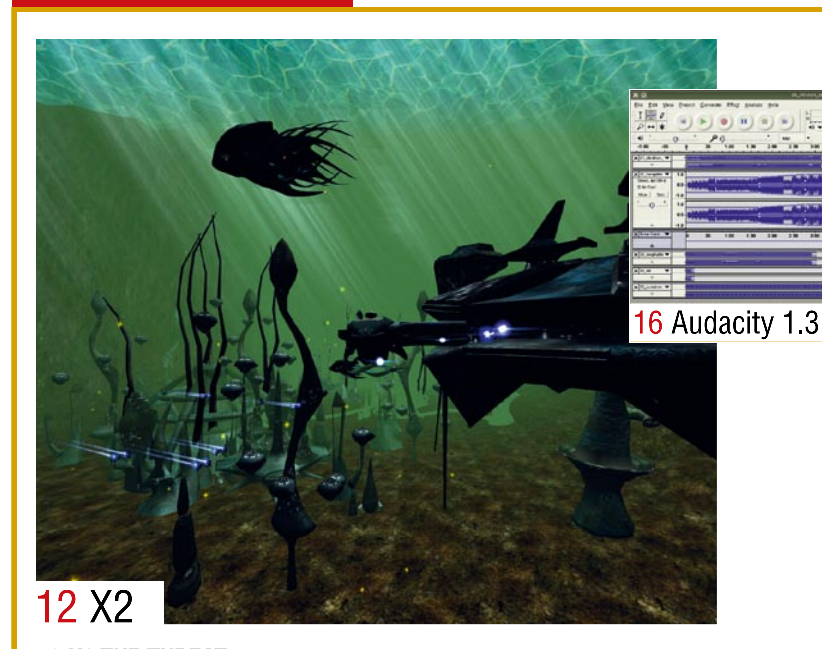
«ОГЛЯНИТЕСЬ ВОКРУГ В ПОИСКАХ АЛЬТЕРНАТИВ.»

Только бедные ли? Или всё-таки «с жиру бесятся»? Стандартная поставка Autocad стоит больше $2 500 и ОС для неё – «прокладка». То же самое можно сказать и про Adobe Photoshop, и про 1С – для рядового пользователя цены совершенно неподъёмные. У меня сильные сомнения, что на LOR общаются одни конструктора, бухгалтеры и работники полиграфии. Действительно, тот же FineReader, судя по дискуссии, используется в основном для сканирования рефератов.

Безусловно, наличие этих и множества других коммерческих программ в Linux – это плюс для всего сообщества. Только оглянитесь вокруг в поисках альтернатив. Оцените свои потребности. Пишите рефераты самостоятельно – мне этот навык сейчас очень кстати.

info@linuxformat.ru

В ЭТОМ НОМЕРЕ...


16 Audacity 1.3

12 X2

НАШ ВЕРДИКТ: ПОЯСНЕНИЕ

Все продукты оцениваются по 10-балльной шкале. 4 обычных параметра оценки: возможности, производительность, простота использования и соотношение «цена/качество», но для свободного ПО последний параметр может быть заменен на наличие или отсутствие документации.

Независимо от набора категорий, мы всегда вычисляем общий рейтинг, подводящий итог нашим высказываниям.

LINUX FORMAT TOP STUFF AWARD

Продукты, выделяющиеся из основной массы, получают престижную награду *Linux Format Top Stuff Award*. Выбираются самые-самые – и только лишь высокая оценка здесь еще ни о чем не говорит.

Для тестирования серверов используется наша собственная разработка, LFXBench 2004, состоящая из четырех основных подсистем: Multi CPU, Single CPU, RAM и Hard Disk. Четыре оценки усредняются и дают общий рейтинг.

1 очко означает, что тестовый компьютер эквивалентен нашему эталону: Pentium 4 1,8ГГц, 512 Мб оперативной памяти и жесткий диск IDE. 2 очка означает, что он вдвое быстрее.

Все тесты выполняются под управлением Red Hat Enterprise Linux 3 AS на соответствующей платформе (x86, AMD64, Itanium). Код компилируется при помощи GCC, если не указано иное.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

MULTI CPU	6.07
SINGLE CPU	3
RAM	2.17
HARD DISK	0.46

ОБЩИЙ РЕЙТИНГ 2.93

КОСМИЧЕСКИЙ СИМУЛЯТОР

X2: The Threat

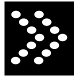


Если вы страдаете по дню, когда Таргоиды будут гонять вас по всей галактике – присоединяйтесь с **Полом Хадсоном** (Paul Hudson) к разведке M5.

САМОЕ ГЛАВНОЕ

Вылитый *Frontier*, но графики больше, а кайфу меньше. Вариант: купить *Frontier* и играть в него на *DOSBox*.

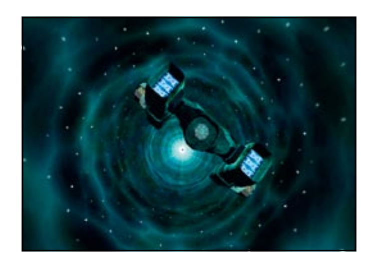
- **РАЗРАБОТЧИК:** Egosoft
- **ПОСТАВЩИК:** Linux Game Publishing
- **САЙТ:** www.linuxgamepublishing.com
- **ЦЕНА:** будет объявлена



Прыгнуть в космический корабль и улететь в космос исследовать звезды было мечтой детства многих умников. Большинство были довольны *Elite* (те, кто постарше) или *Frontier* (помладше): обе игры были безграничными космическими стратегиями, в которых вы могли взять ваш космический корабль, полететь к добрым 2000 космическим системам и торговать с ними в целях наживы. Более агрессивные игроки прихватывали лазерные установки, уничтожали невинных свидетелей потехи ради и зарабатывали деньги на трофеях. Короче, вы могли делать все, что вам заблагорассудится – Вселенная была в ваших руках.

Семь лет назад маленькая немецкая фирма Egosoft выпустила игру под названием *X: Beyond the Frontier* (Вдали от границ), которая воплотила стиль игры *Elite* на современной, аппаратно ускоряемой графикой. *X* завоевала достаточно популярности, чтоб продолжить сериал – вышла *X2: The Threat* (Угроза). Сперва объемы продаж были невелики, но успех пришел сам собой: тысячи пользователей с удовольствием освежали былые славные дни за игрой. Сейчас *X2* перенесена на Linux, благодаря Linux Game Publishing, а значит, мы тоже можем присоединиться к общей массе и вернуться в юность.

Позвольте вам сказать, что *X2* не сумела превзойти *Elite* или *Frontier*. Игровая схема во многом осталась прежней: у вас космический корабль, вы в космосе, везете товары и боретесь с пиратами. Но магией Elite или неповторимым ощущением игры, до сих пор завораживающим сердца людей, которые запускают *DOSBox* и играют во *Frontier*, *X2* явно не обладает.



Скорость света! Новая платная трасса Лондон–Глазго оказалась полна пиратов.



Графика вправду прекрасная; куда ни повернись – глаз радуется.

Людный космос

Многих привлекает в *X2* сложносочиненная игровая Вселенная. Во *Frontier*, вы могли облететь тысячи различных галактик в поисках места для торговли и не встретить никаких признаков жизни, кроме собственной. В *X2*, благодаря массовому прогрессу в аппаратном обеспечении компьютеров, можно встретить тысячи разбросанных по Вселенной космических кораблей и сотни людей и поговорить с ними, и даже функционирует товарная биржа, где цены колеблются по мере продажи или покупки товаров.

В этой вселенной вся информация на английском, кроме почему-то единиц времени, видимо, введенных для оригинальности дизайна. Они довольно остроумно названы, чтобы вас запутать, а именно: сезуры, мизуры, стазуры, тазуры, возуры, мазуры и язуры (sezuras, mizuras, stazuras, tazuras, wozuras, mazuras, jazuras соответственно). Ей-Богу, правда!

Если вы одолеете погрешности номенклатуры, *X2* позволит вам создать галактическую мега-корпорацию, владеющую флотилиями космических кораблей, массивными космическими станциями и прибыльными торговыми путями. Чтобы помочь вам править своей миниатюрной империей, предусмотрено программирование ваших кораблей на автоматическое перемещение по маршрутам, и они набирают вам кредиты, пока вы спите. Зато, увы, гораздо проще выставить время на вашем видеомагнитофоне, чем использовать интерфейс *X2*. Даже самые простые действия требуют хитрых настроек, далеко не всегда очевидных.

Еще более раздражают средства управления, которые не очень-то настроишь, так что приходится уживаться с головоломными параметрами по умолчанию. Чем больше втягиваешься в игру, тем больше осознаешь важность запоминания горячих клавиш для любых действий и нажатий их в строгом порядке для выполнения однообразных задач.



Нечего соваться в панель управления; расслабьтесь и глазейте на лампочки.



Интересно, почему это компьютеры будущего – монохромные?

Командир корабля

Если вы вздумали выбрать путь торгаша, то скоро застрянете. Можете выкинуть из головы любые воспоминания о тщательно прорисованных дисплеях от *Tie Fighter* или *Privateer*: они заменены кое-чем похуже. Гораздо хуже. У каждого корабля свои собственные органы управления, но один практичный принцип господствует во всей игре: забить приборную доску бессмысленно мигающими лампочками, затем спохватиться, что нужны и значимые вещи – вроде питания щита управления – и втиснуть их в скудное оставшееся место.

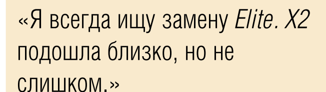
Мало того, большую часть игры вы до ужаса медленно плететесь от звездных ворот (порталов, предназначенных для путешествия от галактики к галактике) к космическим станциям; громадные расстояния в космосе проходятся на невообразимо низких скоростях. Игра делает разве что две уступки, чтобы игрок избежал суицида: особый движок ускорения временного процесса (Singularity Engine Time Accelerator) – устройство, убыстряющее течение времени с коэффициентом, кратным 10, то есть вы затрачиваете на путешествия часы, а не дни; и система автопилота, делающая работу за вас, когда вы идете перекусить.

Но даже это послабление хитроумно подпорчено. Во-первых, автопилот может направляться только к тем объектам, которые вы просканировали с помощью датчика, то есть надо сперва самому слетать к порталу, чтобы автопилот обучился его находить. Во-вторых, иногда вместо посадки корабля на станцию автопилот принимается нарезать над ней круги, и приходится корабль сажать самому, а это раздражающе непросто.

В чем сюжет?

Если вам надоело рвать на себе волосы, продираясь по пути торговца, *X2* предлагает другую опцию: следовать сюжету, который заложили в игру разработчики. Он необычайно слаб, и его не спасает тощий выбор занудных диалогов, которые нужно одолеть, ползя по космосу. За одно из 60-километровых путешествий от ворот до ворот мы вытерпели 5 повторов всего четырех сюжетных линий под преследование картонного вида пиратов, охотящихся за грузом у нас на борту.

Другие игры вставили бы захватывающие ролики для оживления, а в *X2* вместо этого скучные последовательности, генерируемые игровым движком, и полеты космических кораблей проходят под скучную музыку вместо сюжета. В X2 всего130 галактик, куда вы можете полететь, большинство из них принадлежат к одной расе из предусмотренных восьми, но зато имеет уникальные корабли и космические базы. У большинства рас несколько разных кораблей, так что с первого взгляда можно решить, что типы кораблей в игре зависят от вазуры.



ГРЭМ СЧИТАЕТ…

«Я всегда ищу замену *Elite*. *X2* подошла близко, но не слишком.»



Облака, пар из сопла, искры и прочее создают реалистичную атмосферу, ожидающую вашего неспешного исследования.



«Я не слушаю, ля-ля-ля». Эта угловатая дама любезно предлагает вам войти в сюжет. Наш совет – держитесь подальше!

Но дизайн снова выдает свое убожество, как только вы осознаете, что каждый корабль принадлежит к одному из девяти классов: например, M5 – разведчик, M4 – легкий истребитель, M3 – тяжелый истребитель, и различия в рамках класса несущественны. На наш вкус, тут опять-таки недостаток воображения; на худой конец, различные модели выглядят действительно по-разному. Поверите ли, каждая раса имеет свой цвет выхлопных газов: у Аргонов (Argons) он бледно-голубой, у Теладов (Teladi) зеленый, у Хааков (Khaak) пурпурный, и т.д.

Космо-конфетка

Конечно, наиболее заметное отличие *X2* от *Elite/Frontier* – улучшенная графика. Нельзя не признать, что космические эффекты вне корабля красивы. Мимо проносятся частицы, вокруг вздымаются космические облака, а вдали парят недостижимые планеты.

Такой уровень детализации требует довольно мощного компьютера: мы тестировали игру на Athlon 64 3400+ с 256 Мб GeForce 5950 Ultra и 1 ГБ оперативной памяти и добились полноценного качества только ценой замедления (19.395 кадров в секунду при разрешении 1280x1024x32 и отключении сглаживания контурных неровностей и теней на нашей тестовой машине).

Несмотря на юный двухлетний возраст, *X2* смотрится куда дряхлее после приземления на звездную базу, когда ваш персонаж начинает расхаживать вокруг. Успев поиграть в такие игры, как *Unreal Tournaments*, *Postal 2*, *Doom 3* и *Quake 4*, мы привыкли к высококлассной художественной работе, и хотя *X2* достаточно справляется хорошо с грубоватыми углами и резкими границами космических кораблей, прорисовка людей – это провал.

Последняя граница

Единственное, в чем преуспело *X2*, со своим жалким сюжетом, это в возможности построить собственную сложную и бурно развивающуюся экосистему. Беда в том, что ее не с кем разделить – многопользовательский режим в игре отсутствует, и нет ни малейшего подобия онлайновых опций.

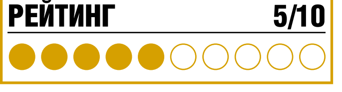
Довольно странно, что фирма Linux Game Publishing выбрала *X2*: под Windows давно уже вышел *X3*, а *X2* был перенесен на Mac около шести месяцев назад. Почему Linux плетется в хвосте? Неужели *X3* показался слишком сложным для переноса? Мы вовсе не хотим выкинуть *X2* из Linux (мы стараемся заполучить любую игру, честно), но нашим лидером она безусловно не станет. Это не *X2*: *Угроза*, это *X2: Угроза заснуть*. LXF

ВЕРДИКТ LINUX FORMAT

ВОЗМОЖНОСТИ	6/10
ПРОИЗВОДИТЕЛЬНОСТЬ	6/10
ИГРОВАЯ ДИНАМИКА	5/10
КПД ВЛОЖЕНИЯ	6/10

По шкале *Elite*, эта игра добилась бы звания Безобидная.

РЕЙТИНГ 5/10



WEB-СЕРВЕР

Apache 2.2



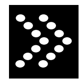


Тяжёлый труд достался 2.0, а сейчас **Дэвиду Каулсону** (David Coulson) понравились мелкие детали.

САМОЕ ГЛАВНОЕ

Модульный сервер для web-сайтов и приложений. Также попробуйте: Boa и *LightTPD*.

- **РАЗРАБОТЧИК:** Apache Software Foundation
- **САЙТ:** www.apache.org
- **ЦЕНА:** Бесплатно, по лицензии Apache



Из Иллинойса пришло много хорошего: Чикагский блюз; Авраам Линкольн; мультик *про Симпсонов*. Но для нас, умников, всего дороже – может, даже краше Ракели Уэлш [американская актриса, – прим. перев.] – мощный web-сервер *Apache*, разработанный Робом Мак-Кулом (Rob McCool) и Национальным центром суперкомпьютерных приложений (National Center for Supercomputer Applications – NCSA). Еще на заре популярности Linux *Apache* стал однозначным выбором web-сервера почти для любых окружений. И небольшие сайты, работающие через телефонное соединение, и огромные кластеры – все строится на *Apache*.

Когда в апреле 2002 года увидел свет *Apache 2.0.35*, это был гигантский шаг вперёд. Однако многие остались на проверенном 1.3, опасаясь болезней роста 2.0. Потребовалось несколько ревизий PHP, чтобы он начала как-то работать, а запуск *mod_perl* был вообще чем-то из разряда черной магии. Из-за этих проблем преимущества нового релиза *Apache* остались незамеченными: большинство дистрибутивов проигнорировали пару релизов 2.0, прежде чем озаботились включением данного пакета.

> «ПРЕИМУЩЕСТВА 2.0 НЕ БЫЛИ ЗАМЕЧЕНЫ ИЗ-ЗА БОЛЕЗНЕЙ РОСТА»

Теперь, получив ещё тепленький *Apache 2.2.0*, люди оглядываются на 2.0, пытаясь разобраться, чего лишились. Крупнейшим изменением было использование многопроцессных модулей (MPM – multi-processing module) для обработки запросов ядром *Apache*. Версии 1.3 использовали структуру на базе процессов (требовался отдельный процесс на каждый запрос), ограничивающую производительность. С выходом 2.0 появился потоковый MPM, известный как «работник (worker)», где один процесс *Apache* распадался на внутренние потоки для обработки HTTP-запросов.

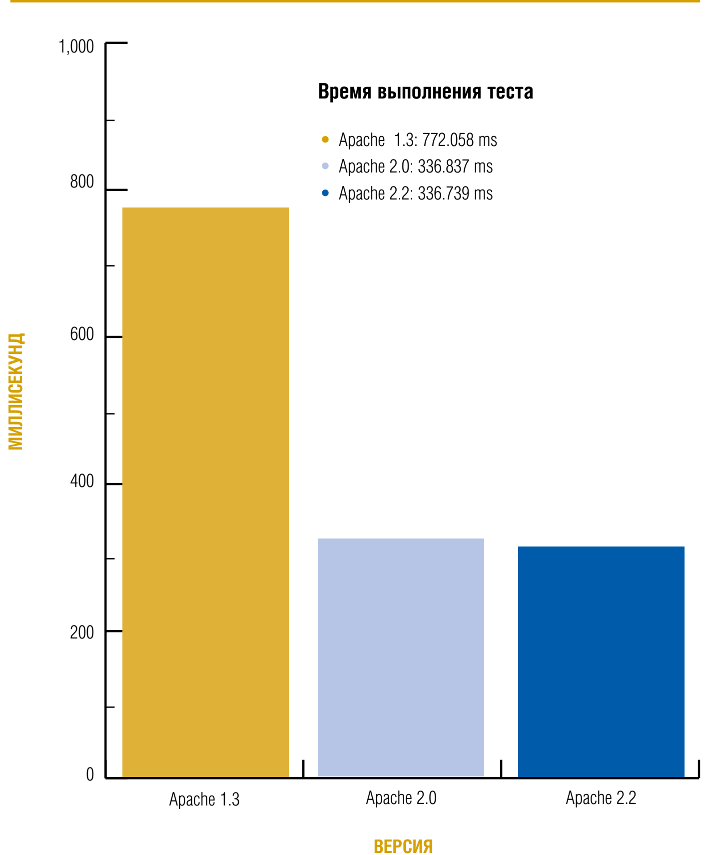
Использование потоков значительно повышает производительность *Apache*, полезную в условиях хостинга: HTTP-демон меньше, а отклик быстрее. Потоки «работника» позволяют обслужить гораздо больше одновременных запросов, поскольку в одном процессе Unix потоков можно запустить неприлично много. По мере захвата потоков активными запросами, отпочковываются дополнительные процессы, создающие новые потоки.

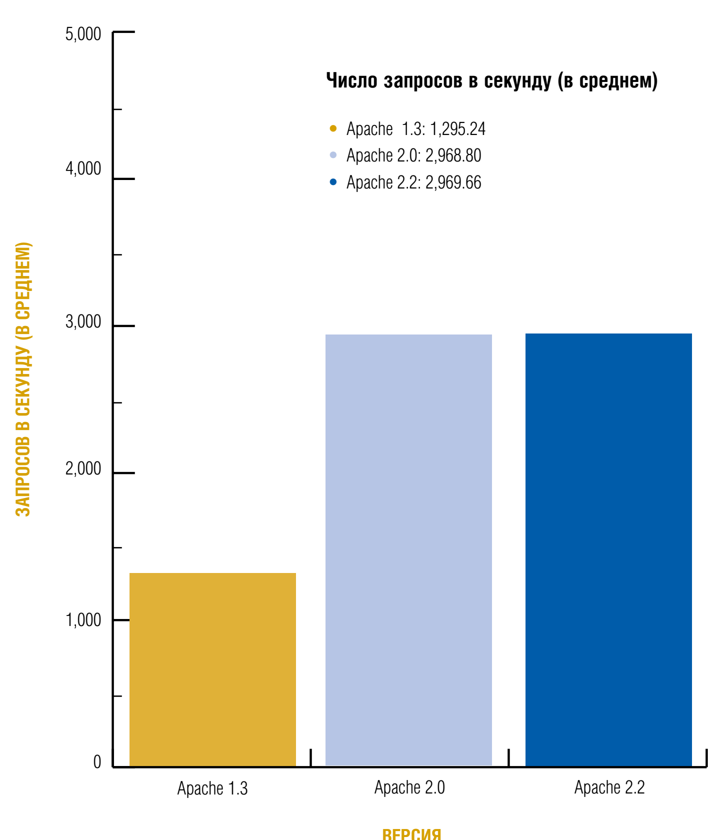
Apache 2.2 предлагает также MPM по имени «perchild» (опять-таки впервые появившийся в 2.0), который умеет отпочковывать процессы *Apache* с различными правами. Это превосходно для виртуального хостинга, потому что сценарии PHP, CGI (Common Gateway Interface) и *mod_perl* могут исполняться от имени конкретного пользователя (а не общего пользователя демона, как в стандартных инсталляциях).

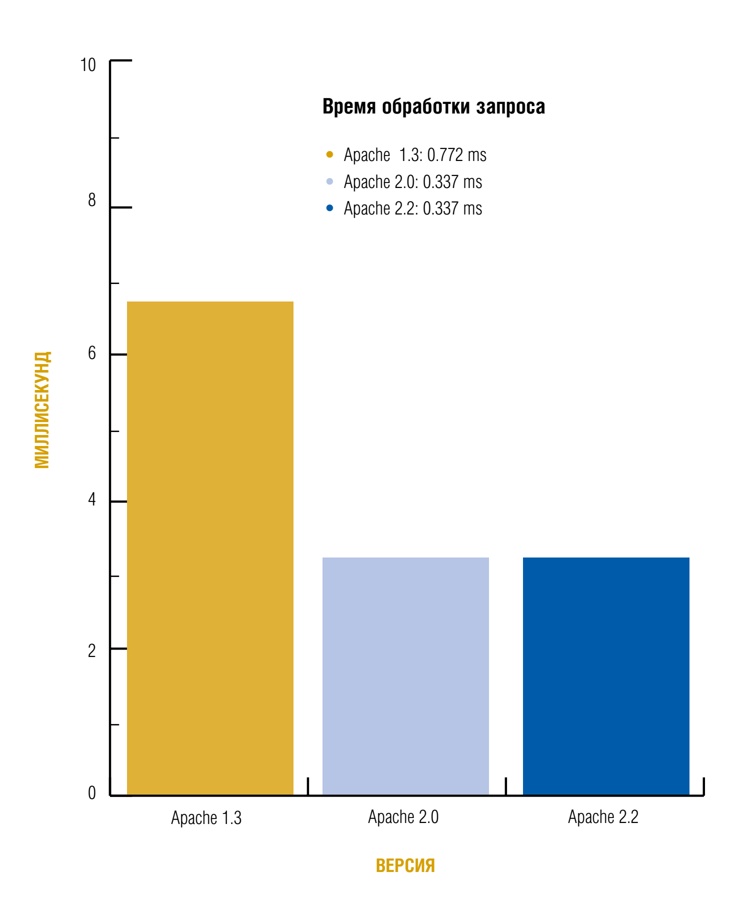
Для динамического виртуального хостинга это не идеально, поскольку пользователей нужно определять в конфигурации *Apache* и на каждый UID требуется свой процесс; но зато возникает дополнительный уровень защиты за счёт «правильного» распределения виртуальных хостов по UID'ам. На внутреннем уровне «prechild» предоставляет те же преимущества потоков, что и «работник», позволяя производительности *Apache* оставить другие web-сервера далеко позади.

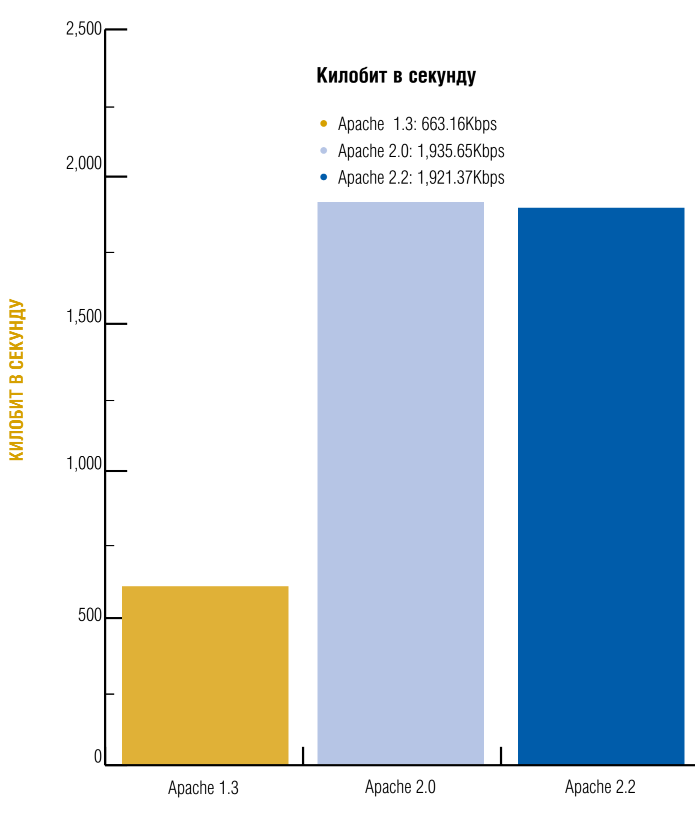
Барьер 2 ГБ

После появления MPM в версии 2.0, для Apache было сложно выпустить 2.2 с такими же эффектными новшествами – возможно, наиболее близким по значимости является снятие бесславного ограничения 2 ГБ на размер файла в 32-битном окружении. На









ТЕСТЫ ПРОИЗВОДИТЕЛЬНОСТИ

Для тестирования производительности новых выпусков, мы запустили *Apache 2.2* вместе с *Apache 1.3* и *Apache 2.0,* в версиях 2.0 и 2.2 использовались потоковые MPM. Тестирование проводилось на стабильной системе Debian с ядром 2.6.15.5, библиотекой *Native POSIX Thread Library* (NPTL) и *Glibc 2.3.2*.

Мы запустили тестовую утилиту *Ab* (входящую в стандартную поставку *Apache*) для испытания наших инсталляций.

Наши тесты исполнялись на системе P4 3.06 ГГц, 2 ГБ ОЗУ и программным RAID-5 на дисках SATA. Мы убедились, что запросы выполняются полностью из ОЗУ, без обращений к диску, и использовали одни и те же файлы для всех инсталляций *Apache*. Чтобы убедиться в чистоте наших экспериментов, мы запускали их 50 раз для каждой версии и усредняли лучшие 75% результатов.

данный барьер гневался член ASF Рич Боуэн (Rich Bowen), озвучив выступление «Почему я ненавижу web-сервер *Apache*» на июльской конференции *ApacheCon*, и последние несколько лет он был предметом забот, поскольку дистрибутивы, выпущенные на DVD, нельзя было распространять через *Apache* и приходилось полагаться на другие HTTP-серверы или *BitTorrent*. Так что разрешение этой проблемы версией 2.2 вызвало всеобщий восторг.

Другие улучшения *Apache в 2.2* связаны с функциями прокси и фильтрацией, что порадует людей, запускающих *Apache* в режиме балансировки нагрузки или для обслуживания больших объёмов статического контента.

По части инсталляции, 2.2 почти полностью идентичен 1.3 или 2.0, только для модулей сторонних производителей, например, PHP, используется *apxs2*, а не *apxs*. MPM можно либо вкомпилировать в *Apache*, либо динамически подгружать как модуль. Сборка 2.2 занимает немного больше времени, но если отключить ненужные модули – выполняется почти мгновенно. Модули можно собирать «на лету», без тотальной переустановки, чем значительно упрощается обновление модулей и тестирование новых функций. Вооружившись хорошо спроектированным API, можно сделать практически всё без перезагрузки web-сервера, хотя, как и в случае с ядром Linux, перезагрузка модулей покамест остается препятствием.

В принципе, последние версии *Apache* загружаются чуть медленнее, чем 1.3. Однако, в зависимости от конфигурации и подгружаемых модулей, время запуска можно существенно сократить.

Просто быстрее других

Первый вопрос любого администратора о развёртывании web-сервера – «Насколько быстро он будет обслуживать мой контент?». *Apache 2.0* значительно превосходит в этом отношении 1.3, особенно при обработке большого числа одновременных запросов. Наши тесты показывают 50% сокращения времени отклика при переходе с 1.3 на 2.0, а 2.2 улучшила 2.0 лишь немного.

Для тестирования на реальной задаче мы шесть месяцев эксплуатировали *Apache* в промышленном режиме на трёхсерверном кластере; двумя из серверов были обновленные до версии 2.0 сервера 1.3. Более ранние инсталляции *Apache* начинали «тормозить» при 250 одновременных запросах, ожидая завершения установленных соединений, прежде чем создавать процесс для обработки новых. Время отклика порядка пяти секунд было обычным явлением. В случае с *Apache 2.0* наш рекорд составил 450 соединений на сервер, с временем отклика кластера около 0,5 секунды. Причем нагрузка на процессор немного снизилась, а производительность возросла. Безусловно, без такой «рабочей лошадки» как *Apache*, мы не достигли бы такой статистики.

Всё ещё фаворит

Для платформ Unix решение использовать *Apache* принимается не задумываясь: он предоставляет невероятную гибкость и производительность, практически не оставляя шансов конкурентам.. Даже для пользователей Windows версия *Apache* для Win32 предоставляет открытый web-сервер, легко интегрирующийся в систему как сервис и работающий много лучше, чем *Microsoft Internet Information Services*.

Всем, кто сейчас использует *Apache* 1.3, советуем убедиться, что все их модули уже применяют новый API 2.x, и немедля приняться за переход к новой версии. Пользователи *Apache 2.0* могут пожелать дождаться пары пересмотров версии, принимая во внимание возможные проблемы с API для сторонних модулей. В перспективе, *Apache 2.3* предоставит дополнительные функции динамического хостинга, наряду с повышением производительности, которое очень понравится владельцам больших серверов.

На данном этапе *Apache* – лучшее приложение Linux. Текущий релиз, как всегда, обеспечивает производительность и стабильность, которую мы привыкли ожидать от столь уважаемого и популярного проекта. Прости нас, Ракель. LXF

ВЕРДИКТ LINUX FORMAT

ВОЗМОЖНОСТИ	10/10
ПРОИЗВОДИТЕЛЬНОСТЬ	10/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ДОКУМЕНТАЦИЯ	10/10

При производительности, ранее наблюдавшейся лишь на небольших HTTP-серверах, версия 2.2 идеальна для развёртывания и больших, и маленьких web-сайтов.





АУДИОРЕДАКТОР

Audacity 1.3

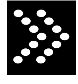
НА ДИСКЕ

Грэм Моррисон (Graham Morrison) исследует последнее обновление почтенного кроссплатформенного аудиоредактора.

САМОЕ ГЛАВНОЕ

Для частного цифрового вещания (podcasts), MP3 и других форматов аудио редактор – один и важнейших инструментов рабочего стола. См. также: *Sweep* или *Rezound*.

- **РАЗРАБОТЧИК:** Доминик Маццони (Dominic Mazzoni)
- **САЙТ:** http://audacity.sourceforge.net
- **ЦЕНА:** Бесплатно по лицензии GPL



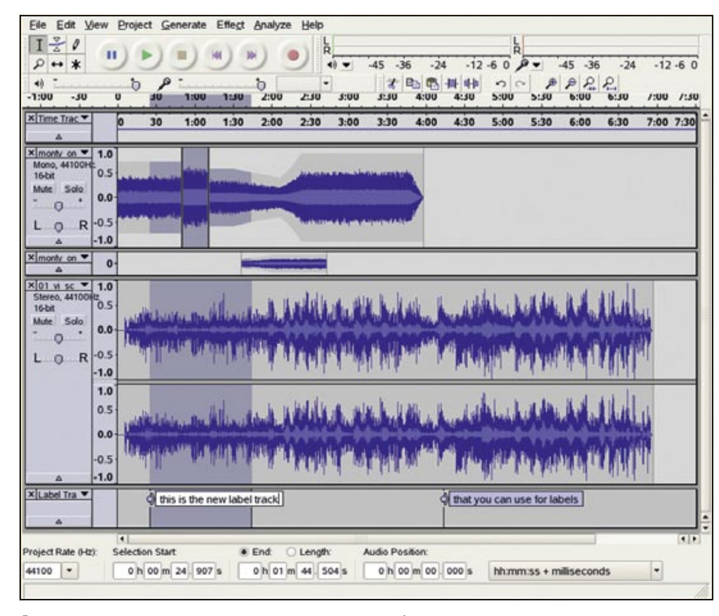
Нам всегда нравился *Audacity*. Этот звуковой редактор буквально набит функциями: его визитная карточка – редактирование нескольких треков и кроссплатформная совместимость, а обработок эффектов тут больше, чем в мастерской BBC – Radiophonic Workshop. Однако, невзирая на репутацию самого документированного аудиоредактора, доступного под Linux, он не победил в Сравнении *LXF69*. Мы посчитали его пользовательский интерфейс немного тяжеловесным, а выходной поток – накладывающим ограничения на использование в связке с другими аудио приложениями, типа *Muse* или *Rosegarden*.

Но мир меняется, и аудиоредакторы сейчас умеют значительно больше, чем просто нарезать на куски радиопередачу или ваш последний шедевр. Для новичков существует мир частного вещания, где каждый может стать звездой. Можно сочинять музыку, сотрудничая через Интернет, и редактировать записи для личного плейера. ПО для подобных задач немножко отличается от компьютерного эквивалента «склеивателя» магнитофонных лент; здесь-то и пригодится *Audacity*.

Версия 1.3 – продукт двухлетней работы после последнего крупного релиза. Объявлено, что в нем ликвидированы сотни ошибок и введены новые существенные возможности, в основном связанные с интерфейсом. Качество звука прежнее,так как *Audacity* уже использовал хранение промежуточных данных в формате 32 бита с плавающей запятой при обработке сэмплов и справлялся с редактированием 24-битовых файлов при частоте дискретизации 96 кГц – куда уж больше.

Несмотря на внешнюю неуклюжесть, старый интерфейс пользователя был весьма разносторонним. Вы могли загружать множество треков в одно окно и регулировать и перемещаться в каждом треке независимо. Текущая версия включила отдельные треки для редактированных «клипов». Клипы – кусочки, которые вырезаются или копируются из выделенного аудиофрагмента, и можно скомбинировать полдюжины или более клипов. Это уже близко к работе профессиональных аудио-приложений типа *WaveLab* от фирмы Steinberg. В Audacity даже присутствует примитивная форма «неразрушающего» редактирования: срез не необратим, его можно откатить, просто щелкнув на линии среза.

Вы можете вырезать или скопировать из одного трека, добавить к проекту новый пустой трек и вставить содержимое буфера. Этот клип перемещается независимо от исходного трека, при помощи инструмента сдвига во времени, который позволяет запросто перетаскивать аудио блоки влево-вправо (умно придумано затенять перемещаемые блоки в выделенном, то есть подсвеченном, диапазоне). Это лучшая возможность *Audacity*, превращающая приложение в отличный инструмент для подстройки сигналов по временным отметкам, например, речи диктора и звуковых эффектов. Если вам требуется детальное редактирование, этот инструмент – для вас.



Аудио-клипы, вместе с амплитудными огибающими, можно двигать вертикально по трекам или горизонтально – по времени.

Вот это да!

Имеется и другое улучшение интерфейса пользователя, идущее рука об руку с множественными клипами: раздвигаемые и расширяемые треки. Если вы понаделали достаточно клипов, кончится тем, что треков у вас будет дюжина или более. Чтобы управиться с ними, можно уменьшить занимаемое ими места, подобно сворачиванию окна в Windows. Место, которое занимают треки, сжимается, но видимая амплитудная информация достаточна для навигации по проекту.

Другие графические улучшения – эффективный трек маркировки, где вы можете поместить ваши примечания или титры, а также трек «оси времени», которая помогает синхронизировать клипы в больших проектах. Прочие улучшения, в основном, глазу невидимы.

Audacity, несомненно, стабилен. Но чувствуется, что разработчики данной версии упустили реальную возможность разобраться с серьезными недостатками. Например, аудиоэффекты все еще производятся не в режиме реального времени. Вместо этого надо выделять область трека и прослушивать ее. Это напоминает темные времена редактирования звука и безусловно должно быть изменено. Та же история и с инструментом спектрального анализа: кому понравится, что можно анализировать только 28,3 секунды записи? Самый же большой изъян *Audacity* – использование библиотеки *portaudio*, то есть выходной аудиопоток прикован к устаревшей системе «OSS», которую давно заменили на ALSA и которая может серьезно осложнить стыковку с другими приложениями. Вы, правда, можете скомпилировать редактор с поддержкой ALSA и Jack, но похоже, что данная поддержка включена как бы вдогонку.

Тем не менее, не отворачивайтесь от *Audacity*: программа позволяет выполнить любое редактирование, только не рассчитывайте, что все будет просто. LXF

Треки можно сужать, ас новым встроенным диалогом можно даже скидывать ваш микс на FTP-сервер.

ВЕРДИКТ LINUX FORMAT

ВОЗМОЖНОСТИ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	6/10
ДОКУМЕНТАЦИЯ	5/10

В Audacity 1.3 имеются стоящие изменения, но звуковой движок требует внимания.

РЕЙТИНГ 7/10

ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ

KDevelop 3.3



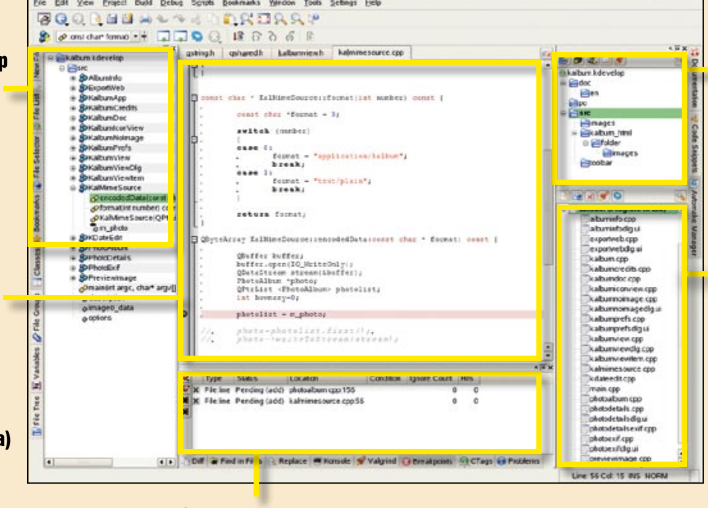
Из-за этой среды разработки KDE у **Грэма Моррисона** (Graham Morrison), забросившего свои проекты, стало одним оправданием меньше.

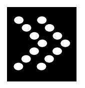
САМОЕ ГЛАВНОЕ

Все, что вам необходимо, чтобы разработать ПО в рамках одного приложения.
См также: *Eclipse* и *Anjuta*..

- **РАЗРАБОТЧИК:** группа разработчиков KDE
- **САЙТ:** www.kdevelop.org
- **ЦЕНА:** бесплатно, под лицензии GPL

KDEVELOP НАВСКИДКУ





Среда разработки KDE развивалась рука об руку с самим рабочим окружением. *KDevelop* был переписан с нуля между выпусками KDE 2 и KDE 3 и получил рабочее название *Gideon*. Вместо вызовов модулей и ограниченного пользовательского интерфейса ранних версий *KDevelop*, *Gideon* отличался полностью открытым пользовательским интерфейсом, основанным на модульной архитектуре, обеспечивающей все необходимое: от встроенного редактора кода до документации и управления проектами. *Gideon* рос вместе с KDE 3, пока не вырос в *KDevelop*.

Сейчас наступает другой переходный период – KDE 3 вскоре сменится KDE 4, и доказательством высокого качества разработки *Gideon'a* является тот факт, что *KDevelop* нужно просто немного расширить для соответствия новой версии *Qt*, которая ляжет в основу нового Клевого рабочего стола. Разработчики могут рассматривать *KDevelop 3.3* как очередную ступеньку. Так как он включен в официальный релиз KDE 3.5 – последнюю версию, где будет использован *Qt 3* – то основной упор был сделан на внедрение поддержки *Qt 4*, но есть и другие значительные добавления, например, улучшенный отладчик (для C, C++ и Ruby) и поддержка PHP 5.

Несмотря на то, что *KDevelop* поставляется вместе с KDE, он не ограничивается рамками *Qt*/KDE. Список поддерживаемых языков постоянно расширяется, и даже до включения PHP 5 вы могли создавать проекты с использованием таких языков, как Ada, Fortran, Perl, Java, Pascal, и даже shell-скриптов. Имеется и хорошая поддержка *GTK*/Gnome, причем во всех перечисленных примерах в качестве стартовой точки предоставлен работающий шаблон, включая «Hello World» для новичков. Это неплохой способ привлечь новых разработчиков, но это также неплохой способ разрабатывать прототипы, особенно когда они используются с формами *Qt Designer*.

Сильно похорошел

Единственное, что сдерживало развитие предыдущих версий *KDevelop* – менеджер проектов. Он является частью интегрированной среды разработки (IDE) и следит за состоянием ваших исходных файлов, наряду с файлами данных, например, с изображениями или другими двоичными данными. Особенно важно для IDE Linux наличие менеджера проектов, берущего на себя некоторые проблемы Makefile и Automake и нормально поддерживающего зависимости..

Проблема заключалась в следующем: в предыдущих версиях даже самые простые задачи вроде импортирования внешнего файла поддерживались плоховато, и часто приходилось редактировать Makefile вручную. Не помогало делу и то, что документация, казалось, была написана невидимыми чернилами.

К счастью, в новой версии управление проектами было реструктурировано. Вместо того, чтобы пытаться справиться со всевозможными средами программирования, менеджер проектов окончательно абстрагировался от основного приложения и опирается лишь на модуль для каждого типа проекта.

Пользовательский интерфейс такой же как и в предыдущей версии (3.2), но благодаря внутренним переменам, он показался нам гораздо эффективнее. Вы можете добавлять и убирать файлы из проекта; меню и кнопки вполне разумны; и *KDevelop* успешно добавляет строки зависимостей в *Makefile*.

Разглядим поближе

Проекты обычно начинаются с использованием мастера, который проводит вас через начальные этапы, например, выбор языка и используемой библиотеки. Главное окно приложения выдержано в стиле KDE, то есть невероятно перегружено. Целых 20 панелей-вкладок окружают основное пространство для редактирования еще до того, как вы начинаете открывать свои файлы. Каждая панель выполняет свою функцию: есть панель для просмотра классов (Class Viewer), буфер для фрагментов кода, отладочная панель и т.д., но они должны быть ограничены опциями конфигурации.

Панель документации – из тех, которые не стоит закрывать. Предыдущие версии страдали от усложненного процесса загрузки API и необходимости распаковки документации перед запуском *htdig*-сервера всего лишь для просмотра вызова функции. Сейчас же надо просто указать область поиска, а *KDevelop* довершит остальное.

KDevelop воплощает дизайн KDE. Иногда создается впечатление, что разработчики погорячились с «интеграцией», но зато приложение наполнено возможностями и функциональностью, и несмотря на сложность освоения, это лучший способ разрабатывать KDE-приложения по нашу сторону командной строки. *KDevelop* также отличная платформой для языков, отличных от C++.

Как только вы привыкнете к *KDevelop*, вы поймете, что это эффективное и мощное средство разработки, будь то ваш первый персональный проект или сотрудничество с еще дюжиной разработчиков через *Subversion*. Проблема одна: имея такую прекрасную IDE, разработчики, давшие самим себе обещание добить тот проект Linux, который давно болтается под рукой, не сумеют придумать никакого оправдания за его нарушение. LXF

ВЕРДИКТ LINUX FORMAT

ВОЗМОЖНОСТИ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	6/10
ДОКУМЕНТАЦИЯ	8/10

Исключительная среда разработки, которая понравится любителям языков программирования.

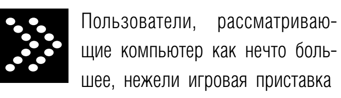
РЕЙТИНГ 8/10




ПРОГРАММЫ ДЛЯ ПЕРЕВОДА

Prawda ли в Linux можно переводить?

Член команды переводчиков *LXF* **Александр Бикмеев** готов поделиться с вами своими наблюдениями.



Пользователи, рассматривающие компьютер как нечто большее, нежели игровая приставка или мультимедиа-центр, рано или поздно сталкиваются с необходимостью перевода текста с одного языка на другой. Как показывают сообщения в форумах, эта тема всплывает достаточно часто.

Существует два подхода к переводу текстов: ручной перевод при помощи обычного или электронного словаря и автоматический перевод при помощи программы-переводчика. Первый предполагает знакомство с языком оригинала, второй – хорошее знание предметной области текста. Как-то Промт перевел мне слово "digg" (кирка) как стоматологический экскаватор, и если бы я не знал, что речь идет о геологе из игры *Settlers*, то ничего бы не понял. Именно из-за таких «шуток» электронных переводчиков, многие знающие язык люди предпочитают пользоваться словарями.

Для ОС Windows существует огромное количество программ как первого, так и второго типа, но в Linux ситуация иная. В связи с этим в форумах часто встречается мнение: «Поставь wine, на него *Promt*, и будь счастлив». Однако мы пойдем другим путем и посмотрим на открытые приложения, работающие в Linux без эмуляции.

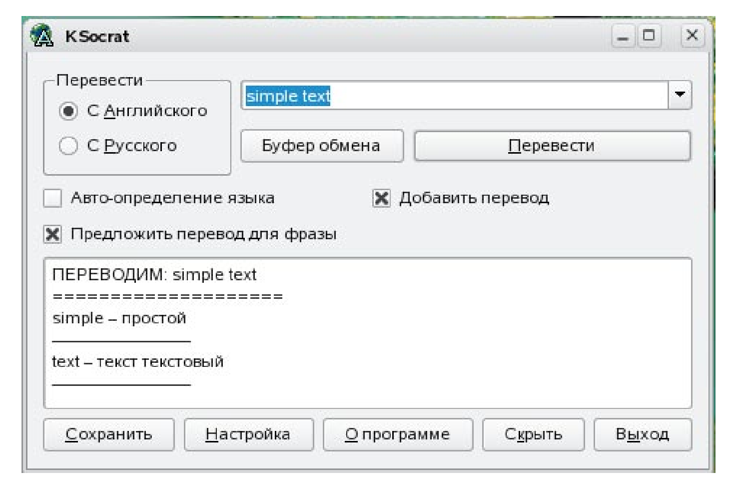
Prawda есть!

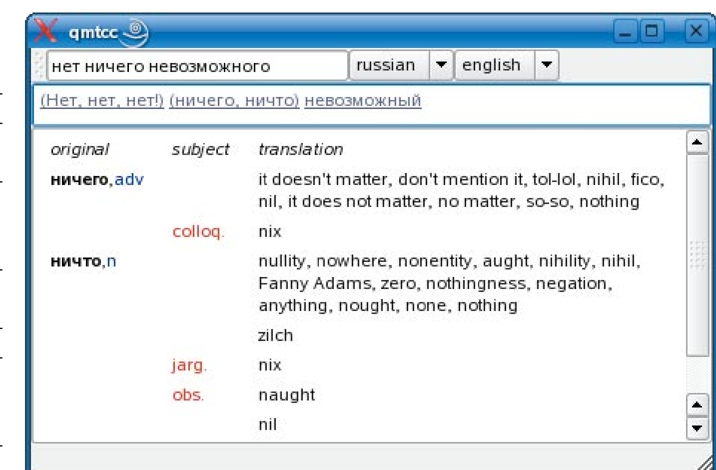
В настоящее время в распоряжении русскоязычных пользователей имеется всего одна программа-переводчик. Это консольная Prawda (Анисимов Д. В., **http://www.prawda.newmail.ru**). При помощи версии 0.4.0 можно перевести тексты с английского и эсперанто на русский. Обратный перевод пока не реализован. В программе используется словарь Мюллера (англ.-рус.) и словарь Бокарева (эсперанто).

Внешний вид программы вполне стандартен и, пожалуй, вызовет ностальгические чувства у ветеранов DOS. Верхнее окно хранит исходный текст, нижнее – перевод. Поскольку программа консольная, то все возможности *Prawda* сосредоточены в текстовом меню, а также доступны по функциональным клавишам.

Программа может показаться неудобной в использовании. К тому же она способна переводить только такие простые предложения, как «You are speaking very well». Кроме того, при переводе не сохраняется форматирование исходного текста. Для того, чтобы сохранить форматирование, необходимо залезть поглубже в меню, вызвать специальный режим, вручную указать границы фраз и абзацев, на что способен не каждый. Кроме того, программа весьма нестабильна при работе с большими или специализированными текстами. Например, при попытке перевода инструкции по установке приложения она «вылетает» без каких-либо объяснений.

Однако, критика – дело хорошее только тогда, когда есть с чем сравнивать. Это единственная программа электронного перевода и само ее наличие внушает оптимизм. Не смотря на ее практически полную бесполезность для рядового пользователя, она может оказаться интересной для разработчиков. Дело в том, что с программой поставляются не только исходные тексты, но и достаточно подробное описание базовой идеи и внутреннего устройства. Более того, автор приглашает всех желающих поучаствовать в разработке. Так что если взяться сообща, то может быть, и выйдет что-то стоящее.





Стоит отметить, что на Западе существует несколько систем машинного перевода для Linux. Это бесплатная система *OpenLogos*, способная осуществлять перевод между английским, французским, немецким, итальянским, португальским и испанским языками и базирующаяся на *PostgreSQL*, а также коммерческая система Terus (**www.terus.co.il**), способная осуществлять перевод с иврита на русский и обратно. На форумах нередко встречаются заявления, что кто-то взялся за портирование *Promt* на Linux, однако результатов пока не видно.

Мюллер и все, все, все...

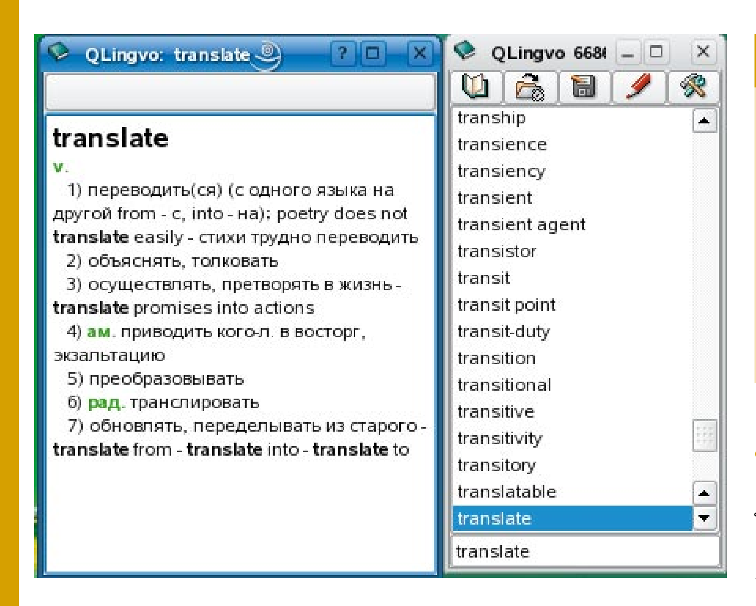
Пользователи Linux в России (да и во всем мире), в основном, более-менее знакомы с английским языком, и потому программы-переводчики с их проблемами им не особо нужны. Однако, эта группа людей остро нуждается в электронных словарях. И здесь Linux, наверное, впереди планеты всей.

В сети существует огромное количество словарей для различных языков. Как и любой файл данных, они имеют некоторый внутренний формат. Наиболее известны следующие форматы и языки разметки электронных словарей: *Mova*, *XDXF*, *Dict*, *Ding* и *DSL*. Соответственно можно найти огромное количество программ, способных читать и использовать словари каждого формата. В Linux особо распространены четыре первых разновидности. *DSL* – это общедоступный язык для описания словарей, разработанный компанией *ABBYY* (**http://www.abbyy.ru**) для системы Lingvo. На сайте **www.lingvoDA.ru** можно найти множество созданных энтузиастами специализированных словарей, написанных «*на DSL*». Перед использованием словаря, сохраненного в формате *DSL*, его необходимо скомпилировать в закрытое внутреннее представление при помощи программы *Dslcomp*, входящей в состав *Lingvo 6* и выше. Поскольку «фирменные» словари Lingvo распространяются в уже откомпилированном виде, воспользоваться ими из-под Linux не удастся. Побродив по сети, можно найти огромное количество других специализированных словарей, а также «стандартный набор»: словарь Мюллера, Даля, Королева и др.

Словари в формате Mova, а также ссылки на ресурсы со словарями и программами, можно взять на сайтах **http://mueller-dic.chat.ru/** и **http://www.mova.org**. Словари в формате DICT на сайте **www.dict.org**. Словари в формате XDXF, а также единственную программу для работу с ними, которую мне удалось найти, можно взять на сайте **atla.sourceforge.net/ru/**.

Теперь, изучив теоретические основы, перейдем к рассмотрению программ, которыми мне довелось воспользоваться.

Первой программой, появившейся на моем компьютере была *Qlingvo* (**http://sanches.boom.ru**). Это электронный словарь, способный открывать словари в формате *Mova*. Интерфейс *Qlingvo* (*см. рисунок*) достаточно прост и напоминает *Lingvo* для Windows. Это окно с панелью инструментов, прокручиваемым списком слов из открытого словаря и строкой ввода текста для поиска. Поиск производится автоматически по мере ввода слова. Таким образом, достаточно набрать несколько букв и слово будет найдено. Имеющаяся на сайте версия 0.24 способна показывать транскрипцию, вос-



А ГДЕ ЖЕ ГИГАНТЫ?

Мы связались с компаниями Promt и Abbyy, чтобы задать вопрос о портировании их продуктов на платформу Linux. Компания Promt сообщила, что в настоящее время не думает о создании Linux-приложения, а компания Abbyy ждет, когда количество просьб о портировании Lingvo на Linux превысит некоторое критическое значение. В то же время на официальном сайте МультиЛекс сказано, что компания взялась за перенос своих словарей в Linux. Кроме того, как следует из документов сайта www.promt.ru, в критически важных лингвистических модулях этой компании используется технология единого кода (single code), что «позволило реализовать решения для ряда программных платформ» в том числе и Linux. Еще в 2000 году компания Promt начала тестирование серверного перводчика для Интернет/Интранет, работающего в Linux, так что надежда есть.

производить звук, то есть произношение слов, а также конвертировать словари из формата Dict в формат Mova и объединять несколько словарей в один. Последняя функция как нельзя кстати, поскольку *Qlingvo* может открыть и использовать только один словарь. Если необходимо найти перевод в другом словаре, то словари легко переключаются, однако проще объединить несколько словарей в один большой и подключать только его. Словари в формате Mova просто копируются в каталог для хранения словарей и становятся доступны из программы без дополнительных действий. *Qlingvo* не идеален, но является, на мой взгляд, прекрасным электронным словарем.

Затем был установлен словарь-переводчик KSocrat (**http://ksocrat.linux.kiev.ua**), использующий словари от одноименного приложения Windows. Словари можно поместить в произвольный каталог, а затем подключить любой из них в настройках программы. Одновременно можно подключить только два словаря: один англо-русский и один русско-английский. Таким образом, *KSocrat* может осуществлять перевод в обе стороны (*см. рисунок*), автоматически определять направление перевода, а также переводить текст буфере обмена. Однако, это не автоматический переводчик, хотя такое мнение иногда встречается. *KSocrat* разбивает фразу на отдельные слова и выдает их перевод, а пользователь должен сам составить из них конечный текст. У данной программы есть существенные недостатки. Например, если слово не найдено в словаре, то оно просто пропускается без каких-либо сообщений. Это относится даже к словам в другом падеже, например, перевод слова сад выводится, а саду – нет. Обнаружена еще и такая ошибка: при вводе достаточно большой строки, которая затем помещается в историю, ширина окна программы автоматически увеличивается до размера наибольшей строки. Избавиться от этого можно, только очистив историю переводов и перезапустив программу. Что самое неприятное, программа не обновляется с 2004 года, поэтому надеяться на скорое исправление недочетов не приходится. Пользоваться данным приложением для перевода больших текстов трудно, но можно, когда нужно получить подстрочник для одной фразы.

Далее была запущена программа *Atlantida* (**http://sourceforge.net/projects/atla**), работающая со словарями в Формате XDXF. Программа написана на Java, представляет собой один файл и для работы требует Java 2 Runtime Environment (JRE или j2re) от Sun Microsystems версии 1.5.0_01 или новее. По заявлению разработчика, программа далека от завершения и в ней еще не реализовано множество задумок, но на данном этапе она способна воспроизводить звук, то есть произношение слов, а также подключать и использовать одновременно множество словарей. При первом запуске программа просит указать каталог, в котором располагаются словари. К сожалению, все мои старания подключить хотя бы один словарь и запустить программу не увенчались успехом. Программа проглатывала имя директории и, повисев немного для приличия в панели задач, исчезала без какого-либо эффекта. Поэтому дать хоть какую-то оценку данной программе я не могу.

Затем настал черед *StarDict* (**http://stardict.sourceforge.net**). Это международный электронный словарь для GNOME, созданный на основе GTK+ и работающий со словарями в формате, немного отличающимся от Dict. Начиная с версии 2.4.5 он устойчиво работает и с KDE. Существует также его консольная версия под названием *SDCV* (**http://sdcv.sourceforge.net**) и сборка под Windows. Следует отметить, что этот словарь входит в такие известные дистрибутивы как *Mandriva*, *AltLinux*, *ASPLinux*. Это самый мощный электронный словарь под Linux, который я использовал. Вот некоторые его возможности:

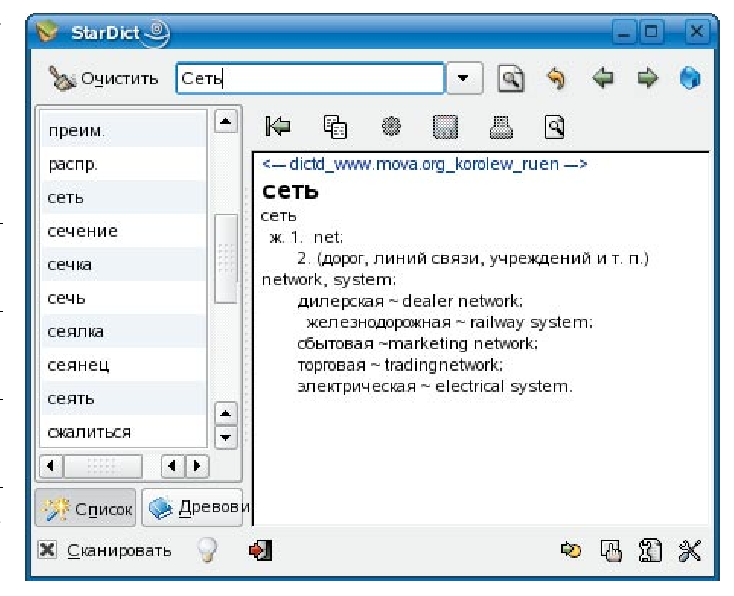
- **Поиск по шаблону**. Пользователь может вводить слова, содержащие «*» и «?», как шаблоны. «*» соответствует любому, возможно пустому, множеству символов, «?» соответствует любому непустому символу. После нажатия ввода, слова, соответствующие шаблону, будут отображены в списке вариантов перевода.
- **Нечеткий запрос**. Если вы не можете вспомнить точное написание слова, то можно воспользоваться «нечетким запросом». Он использует левенштайновский алгоритм для подсчета похожести двух слов, и выдает слова, которые лучше всего введенному запросу. Для использования этой возможности запрос должен начинаться с «/».
- **Сканирование (перевод) выделенного**. После запуска программы, она сворачивается в трей. Затем, при работе с текстом пользователь выделяет мышкой слово, и тут же (или после нажатия настраиваемой клавиши) во всплывающем окне появляется перевод (см. рисунок).
- **Управление словарями**. Пользователь может выключать ненужные словари, а также устанавливать порядок их использования при запросе. Но при этом после каждого изменения требуется перезапуск программы.
- **Произношение**. Если скачать специальный звуковой словарь, то можно не только увидеть перевод слова, но и услышать его произношение.

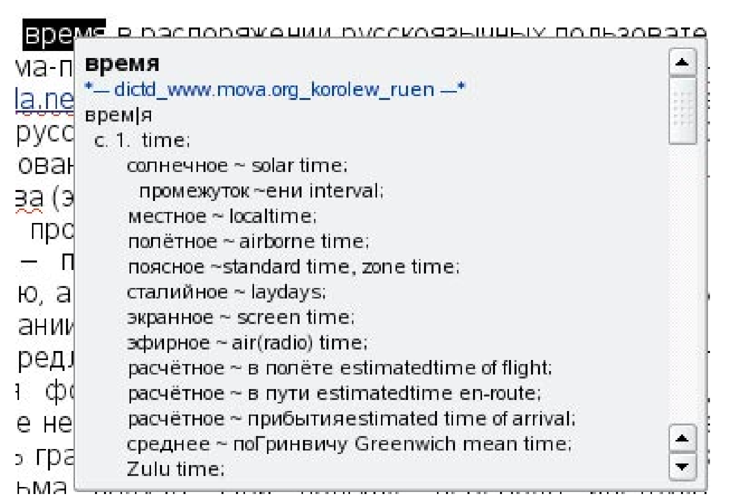
Однако у словаря есть и недостатки:

- В *StarDict* нельзя просмотреть весь список слов, как в *Lingvo* или *Qlingvo*. В окне отображается только небольшой набор слов (около 20), близких по написанию к введенному шаблону.
- Я не нашел ни словаря, ни какого-либо упоминания об отображении транскрипции в *StarDict*.
- Словари размещаются в одной директории, путь к которой пользователь изменить не может.

И все же, учитывая большое число словарей *StarDict* для разных языков, его возможности, простоту установки самого приложения и словарей, а также скорость работы, можно сказать, что *StarDict* является лучшим электронным словарем для Linux. Существует также клон *StarDict* – WiseDict (**http://wise.sf.net/**), ранее называвшийся *stardict-ed*. В нем имеется возможность компиляции DSL-словарей и XDXF-словарей в формат *StarDict*. К моменту, когда писалась статья было доступна версия 0.0.4.

Последним на моем компьютере появился словарь Multitran (**www.multitran.ru**). Поскольку разработка словаря велась Станиславом Иевлевым из *AltLinux*, то найти его можно также и в репозитории





Sisyphus. Установить его было не просто, но помог **www.linuxforum.ru**. Интерфейс очень прост (*см. рисунок*) – строка ввода, рядом с которой находятся кнопки выбора направления перевода, строка, отображающая переводимое в настоящий момент и окно с текстом перевода. Этот словарь является первым портом windows-словаря на Linux. В настоящее время доступна версия 0.0.1. Несмотря на столь раннюю версию, словарь стабилен и весьма порадовал своими возможностями.

Linux-версия *MultiTran* поддерживает перевод между английским и русским языками. Если вводится фраза, то *MultiTran* может разбить ее на составляющие и поместить во вторую строку окна, причем если слово в данной форме не найдено, *MultiTran* предлагает для перевода другие формы слова (*см. снимок экрана*). Не распознанные слова в этой строке отображаются без подчеркивания, а распознанные – подчеркиваются. В скобках могут быть указаны различные однокоренные слова или слова в другом падеже/склонении. Для перевода необходимо щелкнуть на любом из вариантов. В окне с текстом перевода отображается перевод для всех форм слова, которые были указаны в скобках, и при этом часто выдается огромное количество вариантов и синонимов. Для знающих язык это весьма полезно, а для людей слабо знакомых с английским – неудобно. Поиск осуществляется не так быстро, как хотелось бы и паузы могут достигать нескольких секунд. Следует также отметить, что все слова должны быть набраны в нижнем регистре – буквы в верхнем регистре *MultiTran* не распознает.

Может показаться, что недостатков очень много, но достоинства также неоспоримы. Особенно хочется отметить научно-техническую направленность словарей *MultiTran'a*. И поэтому я бы поставил этот электронный словарь на второе место после *StarDict*.

Кроме рассмотренных программ, существует еще множество приложений для работы со словарями. Если вы откроете указанные в статье сайты, то сможете найти ссылки еще десятка на два электронных словарей, многие из которых являются скриптами. Кроме того, можно найти on-line электронные словари и on-line системы машинного перевода. Но, как следует из названия, они работают только при наличии соединения с Интернетом, что для российского пользователя не всегда приемлемо. Таким образом, инструментов для ручного перевода много, но программ-переводчиков для российского линуксоида нет. Может быть, разработчикам все-таки стоит подумать над адаптацией имеющихся продуктов или созданием новых? LXF

СЛОВАРИ ДЛЯ STARDICT

По адресу **ftp://ftp.msiu.ru/education/FSF-Windows/stardict/dicts/stardict-dicts.exe** можно скачать самораспаковывающийся архив (70 Мб), содержащий русско-английские, англо-русские и толковые словари для StarDict. Всего в архиве 37 словарей. В Linux его нужно распаковать в каталог /usr/share/stardict/dic при помощи команды unrar. Словари можно также скачать с сайта **http://gnome.msiu.ru/stardict.php**.

ОПЕРАЦИОННАЯ СИСТЕМА

NetBSD 3.0

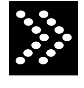


Хотите запустить бесплатную Unix-подобную ОС на вашем тостере? **Майк Сондерс** (Mike Saunders) экзаменует ультра-переносимую операционную систему, применимую даже на кухне.

САМОЕ ГЛАВНОЕ

ОС с открытым исходным кодом семейства BSD. См. также: FreeBSD, OpenBSD и, конечно, Linux!

- **РАЗРАБОТЧИК:** NetBSD Foundation
- **САЙТ:** www.netbsd.org
- **ЦЕНА:** Бесплатно по лицензии BSD

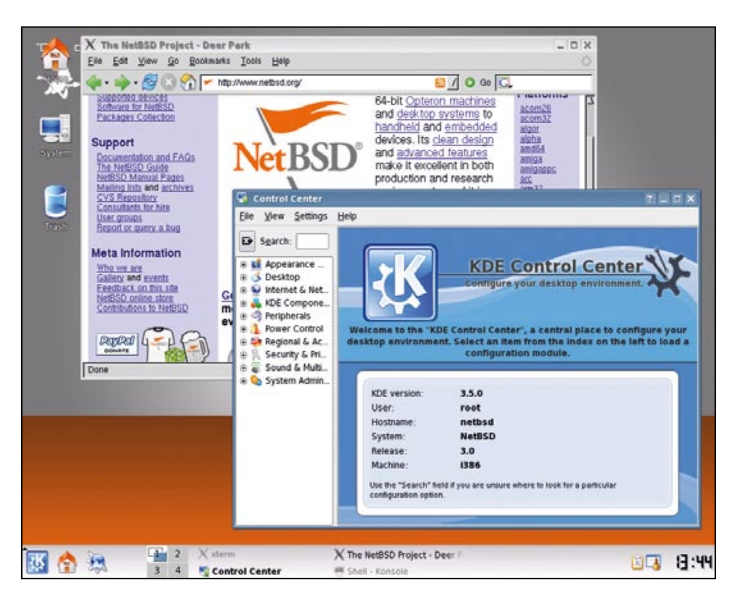


NetBSD – самый незаметный проект из череды ОС семейства BSD. FreeBSD идет в номинации надежной рабочей станции для ПК и серверной ОС, послужной список OpenBSD украшает замечательная безопасность, а Фонд NetBSD корпит в тиши над новыми релизами и поддержкой еще большего количества платформ. Именно изумительная поддержка различных архитектур держит проект NetBSD на плаву – например, релиз 3.0 работает на 57 различных аппаратных архитектурах, включая самые разнотипные машины: обычные ПК, Dreamcast, Atari, Amiga и рабочие станции Sun. Это куда больше, чем предлагают FreeBSD или OpenBSD. Linux работает на множестве платформ, но имеет тенденцию к расщеплению из-за неодинаковых наборов заплаток и деревьев исходных текстов. Невзирая на это, NetBSD разворачивается на всех поддерживаемых машинах из единой кодовой базы.

Следовательно, качество кода, чистота и простота здесь важнее всего. Много кодеров, заинтересованных в изучении ядра, отмечали, как легко разобрать код NetBSD, который имеет постоянных фанатов (особенно в исследовательских и академических кругах), с нетерпением ожидавших NetBSD 3.0.

Sysinst, текстовый инструмент установки NetBSD, не особо изменился и все еще использует простую систему меню для настройки и конфигурирования ОС. Он выглядит достаточно архаично по сравнению с графическими инсталляторами Linux, зато довольно быстр и предоставляет много онлайн-подсказок. NetBSD прославилась скромными системными требованиями, и выпуск 3.0 не стал исключением: из минимальных требований следует, что ОС заработает на компьютере 386-й серии с 4 МБ оперативной памяти и 50 МБ дискового пространства (если, например, вы ставите простой роутер), но для запуска приложений в текстовом режиме лучше обзавестись 16 МБ оперативной памяти и 200 МБ на диске. Для запуска в графическом режиме предпочтительные системные требования уже ближе к Linux: 64 МБ ОЗУ, 1 ГБ места на жестком диске.

Инсталлятор NetBSD выполняет несколько шагов настройки. Вы можете отрегулировать свою конфигурацию после первой загрузки. В типичном для BSD стиле, настройка весьма проста: прочтите man-страницу «afterboot» в руководстве пользователя и установите параметры в **/etc/rc.conf**. Благодаря четкой, хорошо сделанной файловой системе и отлично написанному руководству, полную настройку NetBSD легко уложить в пять минут. Если вы предпочитаете пользоваться окнами и меню, для этой цели в NetBSD предусмотрено средство *Sushi*, оно поможет настроить сеть и пакеты ПО.



Рабочий стол NetBSD далеко не отсталый, на нем самые свежие версии KDE и Gnome, *OpenOffice.org* и *Firefox*.

> «КАК БУДТО СИДИШЬ ЗА ТИПИЧНОЙ МАШИНОЙ LINUX.»

ЧТО НОВОГО В 3.0

- Порт для Iyonix – настольному компьютеру на базе XScale
- Поддержка гипервизора *Xen 2.0*
- Работает на PowerPC 601 (например Mac Performa 6110CD)
- Улучшена поддержка технологии SpeedStep на x86 PC
- Сетевой стек TCP поддерживает сигнатуры MD5 для улучшенной безопасности
- Интегрирован фильтр пакетов OpenBSD
- Поддержка беспроводных сетевых чипов Intel 2100/2200AG/2915 ABG
- Ускорение работы файловой системы FreeBSD за счет опции DIRHASH
- Поддержка подключаемых модулей аутентификации – PAM (Pluggable Authentication Modules)

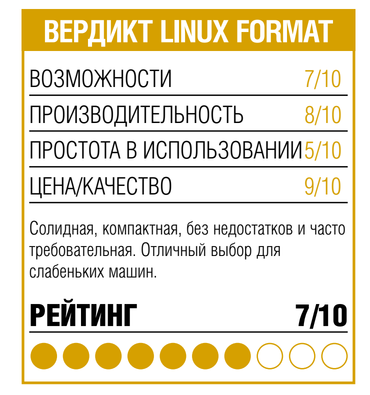
Чистый движок

В использовании, NetBSD выглядит и ведет себя как любая другая ОС на базе Unix и POSIX: вы можете инсталлировать *Bash*, *X*, KDE/Gnome и *Firefox* совершенно так же, как сидя за типичной Linux-машиной. Различия проявляются в разработке и администрировании – а внутри ОС на логичность NetBSD приятно посмотреть. Ядро, библиотеки, документация и средства поддержки разработаны в рамках единого проекта и прекрасно стыкуются, контрастируя в этом плане с Linux, который вследствие неоднородной разработки имеет склонность к разболтанности (пожалуйста, не перепутайте, на кого гневаться).

Pkgsrc – «пакетная система NetBSD» – основана на исходных кодах, а-ля Ports во FreeBSD и *Portage* в Gentoo, и на удивление современна: среди главных пакетов – KDE 3.5, Gnome 2.12.1, *Apache 2.0.55* и *Samba 3.0.20*. Набор инструментов построен вокруг GCC 3.3.3. NetBSD щеголяет бинарной совместимостью с другими Unix-подобными системами, включая Solaris и Linux. На практике это значит, что можно запускать Linux-приложения типа *RealPlayer* и *Quake 3* на x86 NetBSD, используя исходные библиотеки Linux.

Разумеется, NetBSB еще не соперничает с Linux как универсальная настольная ОС: ей требуется более долгое время на обеспечение поддержки самоновейшего аппаратного обеспечения и солидные знания в Unix по части конфигурирования и администрирования. Однако стабильность и безопасность она обеспечивает первоклассную, а в качестве пособия по изучению работы Unix-подобных систем не имеет равных. (И, кстати, мы не шутили по поводу тостера: **www.embeddedarm.com/news/netbsd_toaster.htm)**

• Для подробного ознакомления с BSD-системами читайте материалы на страницах 44. LXF



ВЕРДИКТ LINUX FORMAT

ВОЗМОЖНОСТИ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
ПРОСТОТА В ИСПОЛЬЗОВАНИИ	5/10
ЦЕНА/КАЧЕСТВО	9/10

Солидная, компактная, без недостатков и часто требовательная. Отличный выбор для слабеньких машин.

РЕЙТИНГ 7/10

ЛУЧШАЯ СТАТЬЯ О LINUX И ФРИЛАНСЕ

В период с 20 января по 25 февраля журнал Linux Format совместно с порталом Free-lance.ru проводил конкурс на лучшую статью, посвященную Linux/Open Source и удаленному выполнению работ (фрилансу). Жанр статьи специально не оговаривался: это могла быть история успеха, практическое руководство в стиле HOWTO – все зависело от воли автора. Честно говоря, мы ожидали увидеть материалы от «технарей», разрабатывающих программы для богатых компаний в солнечной Бургундии или удаленно администрирующих сервера, но оказалось, что наш конкурс нашел отклик в сердцах самых разных людей – программистов, администраторов, художников и поэтов в самом широком смысле этого слова. Спасибо всем, кто прислал нам свои работы!

Выбрать победителя, как обычно, было нелегко. Однако, после жарких споров и введения хитроумной системы ранжирования, компетентному жюри все же удалось прийти к единому мнению. Первое место за оригинальность выражения и творческий подход к делу было присуждено **Безносу Юрию** (zhiz0id@gmail.com), автору статьи «Свободный полет». Он лишь незначительно опередил **Никиту Щекина** (parusnik@mail.ru) и его «Путь «от мало до велико». Третье место досталось **Андрею Роговскому** (esupport@esupport.org.ua) за выдающиеся успехи в области удаленного администрирования серверов.

Мы от души поздравляем всех победителей конкурса! Со работами лауреатов можно познакомиться в Виртуальной библиотеке Линуксцентра (http://www.linuxcenter.ru/lib) в разделе «Linux наступает!» и на сайте Linux Format (www.linuxformat.ru) А сейчас слово предоставляется победителю конкурса!

СВОБОДНЫЙ ПОЛЁТ

Когда-то давно я занимался парашютным спортом. Сразу скажу, что прыгать не прыгал, но пропитаться парашютной романтикой успел. За всё время общения с парашютистами ни разу не встречал того, кто бы не мечтал выполнить или повторить затяжной прыжок. «Затяжным прыжком» парашютисты называют прыжок с задержкой до открытия основного парашюта. Расспрашивая о впечатлениях, я замечал, как ребята менялись на глазах. Они рассказывали об этом как о нирване, как о самом классном ощущении, которое они испытывали. Те, кто прыгал, очень хотели повторить, те же, кто не прыгал, от таких рассказов загорались ещё больше.

Но есть и свободный полёт другого рода: свободные наёмники, так называемый «freelance». В то время, как многие из нас привыкли к навязанным стандартам (школа, институт, завод), часть людей работает свободно, без жёсткого графика, так, как им удобно, с желаемой нагрузкой и планируемой отдачей. Эти люди выбирают задачи, которые им предстоит выполнять – сами. Развиваются в том направлении, в котором хотят и, что самое важное, происходит это быстро и эффективно. Основные инструменты фрилансера – это компьютер и Интернет. По сути щит и меч. Но пустой компьютер – просто счётная машинка, а ведь на ней надо что-то обсчитывать. Программы. Программы???

Есть ещё один вид свободного полёта – написание программ с открытым исходным кодом. И более того, с лицензией на свободное использование этого кода. «Ничего не понимаю, – скажут некоторые из вас, – зачем мне всё это?» Для большинства людей всё достаточно просто. Вы можете использовать эти программы для себя и своей работы, не платя никому ни цента. И, что самое важное – это будет правомерно! Тотчас в голове всплывает название операционной среды с открытым исходным кодом – «Linux». По правде говоря, Linux – не операционная среда (да, вот уже вижу, как в меня полетели камни). Linux – это ядро системы, по сути, её сердце. Операционной средой же можно назвать дистрибутив. Дистрибутив – уже полноценная операционная система с изобилием программных пакетов. Как дистрибутивов, так и пакетов к ним сейчас уже тьма, и времена дикой пустынности давно прошли. Мир Linux – это мир энергично развивающегося, современного и первосортного кода. Такие проекты, как *Gnome, KDE, Blender, Inkscape, Gimp* частенько попадаются на глаза и тут, и там. Качественных проектов действительно много.

Но самое интересное, что вы можете поучаствовать практически в любом из них. Большинство встреченных мною Open Source проектов искали помощников. Это не только программисты, это и художники, и переводчики, и корректоры, и тестеры, и фотографы, и архитекторы, и физики, и математики, и масса других. Зачем, казалось бы, помогать кому-либо, тратя своё время на чужой проект? Но на эту ситуацию нужно посмотреть с другой стороны. Можно получить опыт работы в той или иной области практически с нуля. Научиться работать в команде. Использовать эффективные техники работы и многое другое. А с уже полученным опытом найти качественную и достойную работу как фрилансером, так и не фрилансером значительно проще. Ведь что расскажет о человеке красноречивей, чем уже сделанная качественная работа?

Несколько минут длится самый длинный затяжной прыжок до открытия парашюта. Несколько минут. Freelance же может длиться всю жизнь. Всё необходимое есть. Компьютер и подключение к Интернету есть у многих. Программ с открытым исходным кодом достаточно. Проектов, нуждающихся в поддержке – масса. Нужно лишь знать об этом мире. Эта статья – просто весточка.

Набравшись сил и энергии, заточив свой меч и отполировав до блеска щит вы вступите в эпоху борьбы за деньги и коммерческие проекты. Последнее, возможно самое важное – точка старта.

Портал www.free-lance.ru – это отличный способ начать свою фриланс-карьеру. Не единственный, но один из лучших отечественных порталов по найму фрилансеров.

ТАК ЧЕГО ЖЕ ВЫ ЖДЁТЕ?

DISTROWATCH – смотр дистрибутивов

Новости разработки Linux-дистрибутивов за этот месяц.

КАК ЧАСЫ



Ладислав Боднар (Ladislav Bodnar) – основатель, хранитель, начальник сотрудник DistroWatch.com.

Когда говорят о планировании выпусков Linux-дистрибутивов, можно выделить три подхода. Первый – это благоговейно соблюдаемый жёсткий план разработки. Второй – анонс примерного, потом обнаружение, что соблюсти его невозможно, и задержка выпуска, иногда на несколько недель. Третий – вообще полное отсутствие анонсов и планов, когда работа ведётся в традиционном для Open Source ключе: «выпустим, когда сделаем».

«ЧЕТКИЙ ПЛАН ВЫПУСКОВ СТАЛ ГОРАЗДО ВАЖНЕЕ.»

Какой дистрибутив к какой категории принадлежит? Из событий 2005 года ясно, что Ubuntu и новый проект OpenSUSE придерживаются политики публикации – и чёткого соблюдения – своих рабочих планов. Fedora и Mandriva, напротив, обнародуют лишь предварительные планы, которые, судя по задержкам появления beta-версий и финальных релизов обоих дистрибутивов, могут слабо соответствовать действительности. А Debian и Slackware? Эти вообще не утруждают себя планированием, ограничиваясь декларацией общих целей проектирования.

Имеет ли это значение? Что ж, жёсткое планирование выпусков становится всё более и более важным. Во-первых, это повышает доверие к дистрибутиву. Во-вторых, разработка Linux продолжается в бешеном темпе, и многие пользователи хотят знать, когда же появится следующий выпуск, со всеми последними новинками. И, наконец, жесткий план (и политика честного оповещения) избавляют почтовые рассылки и форумы от постоянных вопросов про дату очередного релиза, высвобождая время для обсуждения реальных проблем.

Mandriva: намек на новые приобретения

Увидим ли мы Mandrivaski в 2006 году ?



2005 год был удачным для Mandriva. Хотя их флагманский продукт часто рассматривается как ориентированный на любителей и домашних пользователей, эта парижская фирма в 2005 году продемонстрировала серьёзные намерения и в отношении корпоративного рынка, по крайней мере, у себя дома. Новые заказы поступили не только от крупных фирм вроде French Telecom, но и от правительственных учреждений. Министерство Франции по транспорту, оборудованию, туризму и мореплаванию объявила тендеры на 65000 рабочих станций, причём среди условий – использование на них Mandriva Linux и Windows. Как сказал Франсуа Бансильон (Francois Bancilhon), исполнительный директор (CEO) компании Mandriva, в своем недавнем интервью для Yahoo France, это открывает перед фирмой огромные перспективы в наступившем году.

Учитывая, что прежнюю фирму Mandrakesoft в 2003 году записали в банкроты, основатели Mandriva с трудом верят в происходящее. Хотя приобретение бразильского дистрибьютора Conectiva в прошлом году обошлось компании в сумму порядка 1,8 миллионов евро, его плоды уже начали созревать. Благодаря договору с Positivo, ведущим бразильским сборщиком ПК, Mandriva получает теперь неплохой доход из крупнейшей страны Южной Америки. По словам Бансильона, 20% прибыли Mandriva в 2005 году получено из Бразилии, тогда как из Франции – 25% , из остальных стран Европы – 25% и из Северной Америки 30%.

Исполнительный директор также защищал решение фирмы распространять закрытое ПО в составе коммерческих пакетов своего дистрибутива, в ответ на вопрос о недавнем включении в дистрибутив *Skype* – популярного кроссплатформенного приложения интернет-телефонии. Эта тенденция встревожила некоторых пользователей: «Mandriva всё больше отдаляется от духа Linux», – написал один их них на Infos-du-Net.com. Но Бансильон доволен. Согласно договору с eBay, получившим права на Skype в сентябре прошлого года, Mandriva будет получать комиссионные за каждого нового пользователя, подписавшегося на Skype из Mandriva Linux.



Mandriva отстояла включение *Skype* в дистрибутив. Другое проприетарное ПО в дистрибутиве – *Macromedia Flash*, *RealPlayer* и драйверы Nvidia.



Франсуа Бансильон намекнул, что фирма целит на Россию и Китай.

Бансильон также поведал, что после поглощения Conectiva и Lycoris в 2005 году, Mandriva собирается продолжать политику приобретений и в 2006 году. Он намекнул, что фирма рассматривает Россию и Китай как наиболее перспективные рынки для Linux. Не исключено, что Mandriva уже ведёт переговоры с определёнными Linux-компаниями в этих странах (Alt Linux начинался как локализованная версия Mandrake Linux), так что новости уже не за горами.

Но независимо от исходов этих переговоров и планов, мы уже являемся свидетелями резкого поворота в судьбе фирмы, которая всего три года назад была в глубоком финансовом кризисе. Так держать, Mandriva!

По-Gentooльменски

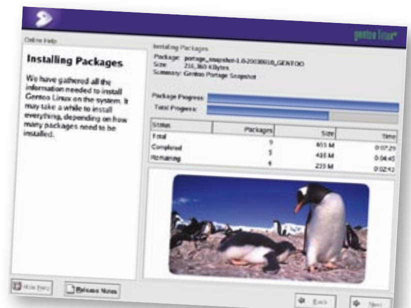
VLOS, RR4 Linux, Kororaa – простые дистрибутивы на базе Gentoo



Gentoo Linux, любимый дистрибутив квалифицированных пользователей, на сегодняшний день – наиболее гибкая и настраиваемая операционная система. Однако появились признаки снижения его популярности: многие пользователи, некогда бывшие в восторге от его гибкости и мощи, считают нудными долгий (хотя и поучительный) процесс установки и бесконечную компиляцию. Три проекта, обратившиеся к данной проблеме, заслуживают внимания.

VLOS – автономный дистрибутив, использующий порт от проверенного временем и Red Hat инсталлятора Anaconda для загрузки набора скомпилированных приложений на ваш жёсткий диск. По завершении установки, VLOS фактически превращается в Gentoo – может, не так хорошо оптимизированный под ваше оборудование, но это легко поправимо перекомпиляцией системы. Некоторые из последних релизов VLOS не обошлись без ошибок, но данный проект обещает много.
http://desktop.vidalinux.com/

Основанный на Gentoo Live DVD RR4 Linux предоставляет огромную коллекцию приложений на загрузочном диске, и многие считают его прекрасной альтернативой Knoppix и другим популярным Live CD. Лучшая часть RR4 Linux – официальный Gentoo Installer, который упрощает установку дистрибутива на жёсткий диск. Хотя и инсталлятор, и Live DVD пока находятся в стадии разработки и могут содержать ошибки, этот проект довольно широко распространён. Существует и 64-битная версия дистрибутива – RR64.
http://www.lxnaydesign.net/



Установка Gentoo при помощи Anaconda, одного из популярнейших инсталляторов для Linux.

Kororaa – ещё одна попытка упростить установку Gentoo на жёсткий диск. Текстовый инсталлятор придётся по вкусу не всем, но наиболее свежий релиз, по слухам, работает замечательно, и пост-инсталляционная настройка позволяет вам настроить систему без особых усилий. Правда, для установки потребуются два CD – один с базовой системой для x86 и AMD64, а другой с графическими компонентами дистрибутива (KDE либо Gnome).
http://www.kororaa.org/

ДИСТРИБУТИВЫ-ЛИДЕРЫ

10 самых посещаемых страниц на DistroWatch.com в ноябре-2005
(среднее число посещений за сутки)

1	Ubuntu Linux	2177
2	SUSE Linux	1358
3	Mandriva Linux	1262
4	Fedora Core	930
5	Mepis Linux	875
6	Damn Small Linux	759
7	Debian GNU/Linux	731
8	Knoppix	594
9	Kanotix	536
10	Slackware Linux	524

DistroWatch.com отслеживает популярность дистрибутивов, основываясь на числе посещений посвященных им сайтов. Хотя оно не отражает реального количества инсталляций, но показывает, какие дистрибутивы привлекают внимание в определённый период времени.

НЕ УСТАРЕЛ ЛИ ВАШ ДИСТРИБУТИВ?

Если только вы не эксплуатируете Linux на сервере (и живёте не в дремучем лесу), то, скорее всего, интересуетесь всеми основными выпусками приложений. Но, кроме случаев дистрибутива на базе исходного кода, типа Gentoo или ветви разработчика одного из дистрибутивов с открытой моделью разработки, вроде Mandriva Cooker или OpenSUSE, наслаждение последними пакетами придётся отложить до выхода следующего стабильного релиза вашего дистрибутива. Насколько «свежи» основные дистрибутивы? Предлагаем сравнительную таблицу основных компонентов.

Дистрибутив	Ядро	Gnome	KDE	OpenOffice.org
Debian 3.1	2.4.27	2.8.1	3.3.2	1.1.3
Fedora Core 4	2.6.11	2.10.0	3.4.0	1.9.104
Mandriva Linux 2006	2.6.12	2.10.1	3.4.2	1.1.5rc4
Slackware 10.2	2.4.31	-	3.4.2	-
SUSE 10.0	2.6.13	2.12.0	3.4.2	1.9.125

Лис-машина



Fox Desktop Linux 1 – серьёзный подход к управлению программами



После более чем года разработки, на серверах стал доступен для скачивания первый стабильный релиз Fox Desktop Linux. Новый дистрибутив основан на Fedora Core 4 и понравится начинающим линуксоидам и низкоквалифицированным пользователям. Кроме приятной темы KDE, похожей на интерфейс пользователя Mac OS X, итальянские разработчики немало потрудились над предоставлением удобного центра управления для различных настроек операционной системы.

Ряд наиболее интересных модулей Центра Управления спрятаны под вкладкой управления пакетами. Fox использует Smart (первоначально разработанный в Conectiva) – новый и многообещающий способ выполнять установку и удаление пакетов в дистрибутивах Linux. Менеджер пакетов имеет интуитивно понятный графический интерфейс и иконку на системной панели для уведомления пользователей о доступных обновлениях.

Другая интересная функция в Fox Desktop Linux – наличие нескольких модулей и опций конфигурации, связанных с безопасностью. Будь то межсетевой экран, шифрование файловой системы или параметры аутентификации, всё настраивается сравнительно просто с помощью комфортных и аккуратных графических приложений. Многие «родные» для KDE конфигурационные модули можно вызвать из Центра Управления Fox, превращающим данное приложение в своеобразный командный пункт для администрирования системы.



Это Mac OS X или KDE? На Fox Desktop разницы не чувствуешь…

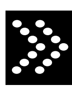
Хотя это первый официальный релиз проекта, разработчики уже продемонстрировали полное понимание рынка дружественных к пользователю операционных систем Linux и способность разработать уникальный продукт. Fox Desktop стоит попробовать, особенно если вы новичок в Linux и любите Mac.
http://www.foxlinux.org/ LXF

СРАВНЕНИЕ

Каждый месяц мы изучаем тонны программ, чтобы вам не пришлось заниматься этим самостоятельно!

АСТРОНОМИЧЕСКИЕ ПРОГРАММЫ

Грэм Моррисон мог бы работать на желтую прессу, но он наблюдает за звездами.



Linux не испытывает недостатка в астрономических приложениях. Они охватывают диапазон от небольших академических утилит командной строки для поиска двойных звездных систем до огромных графических мегалитов, стремящихся втиснуть целую Вселенную на ваш рабочий стол. Почему их так много? Потому что Linux, благодаря UNIX-корням, получил в наследство огромную армию технически подкованных разработчиков, которые, так уж получилось, живо интересовались астрономией. Это важно: чтобы создать компьютерное приложение, являющееся сплавом математики и физики, необходимо обладать серьезными техническими знаниями.

Мы пошли по практическому пути, ограничившись приложениями, небесполезными и для нас, простых смертных. Обычно это означает настольный планетарий – средство моделирования ночного неба, которое вы можете увидеть из своего сада, если, конечно, дождя нет, а сад есть.

Настольные планетарии – прекрасное средство для изучения звезд и отличная гимнастика для глаз, вы можете увидеть, как выглядело небо в прошлом, или как оно будет выглядеть в будущем. Во многих из них рассчитываются фазы Луны, фазы планет и их спутников, и даже орбиты комет и астероидов. Мы также рассмотрели астрономические приложения, предоставляющие более широкие возможности. Одно из них, например, позволяет осмотреть поверхность Марса, а другое – путешествовать по Млечному пути со скоростью, во много раз превышающей скорость света.

Лидером нашего обзора должна была стать программа, покорившая наше воображение и вдобавок представившая небесное шоу над нашими головами на хорошем научном уровне. Мы также учитывали интерфейс, качество графики и размер звездной базы данных приложения. Приятно, что независимо от нашего вердикта все приложения, рассмотренные в обзоре этого месяца, предлагают нечто полезное, и многие, возможно, вдохновят вас стряхнуть пыль со старого телескопа и провести с ним ночью несколько часов.

УЧАСТНИКИ СРАВНЕНИЯ

KStars



Классический настольный планетарий.

• **ВЕРСИЯ:** 1.2.0 • **WEB:** http://edu.kde.org/kstars • **Цена:** Бесплатно по лицензии GPL



Итак, у вас есть телескоп, термос с кофе и сердечный друг, чтобы составить вам компанию. Возможно, вы захотите добавить к этому списку и ноутбук с установленным *KStars* – *KStars* набит функциями и является прекрасным подспорьем, когда надо распознать звезду.

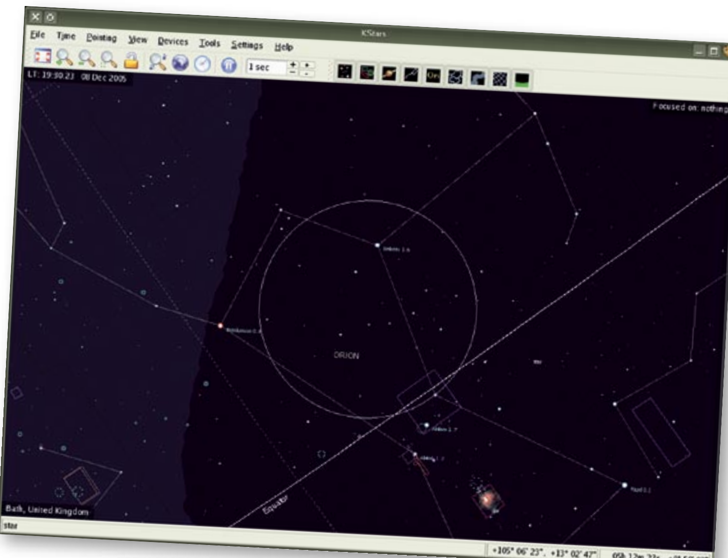
Еще не так давно компьютерные настольные планетарии обновляли экран мучительно долго. Параметры обзора, такие как направление и угол наблюдения (в астрономии – «склонение»), необходимо было сразу выбирать безошибочно, чтобы дождаться результата не зря. Но *KStars* – представитель нового поколения приложений – использует преимущества современных процессоров для отображения изменений небесной сферы в реальном времени.

После запуска программы, например, вашему взору предстает южная часть неба для вашего географического местоположения. Даже если вы сейчас загораете на солнышке, *KStarts* отобразит звезды так, как если бы оно давно ушло за горизонт. Кружки, изображающие звезды, окрашены в цвета, соответствующие их реальным спектральным данным, а размер кружка характеризует относительную яркость звезды. И все это на иссиня-черном фоне.

Отображение звезд весьма эффектно и реалистично, по крайней мере если сравнить его с фотографией и с тем, что вы видите невооруженным глазом. При необходимости постоянного обновления (например, когда вы перемещаетесь по небесной сфере) количество отображаемых звезд уменьшается до нескольких наиболее важных. Тут может возникать проблема: если вы движетесь внутри скопления малых звезд, обновление неба приводит к их исчезновению, и это затрудняет навигацию. *KStars* будет работать лучше, если принимать во внимание только небольшое количество звезд, отображаемых на экране.

Что я вижу?

Изображение ясное и схематичное: метки объектов – наиболее ценное качество *KStars*. Метки замечательно выглядят, безупречно отображаются и грамотно расположены. При движении по небу все наиболее важные небесные тела перемещаются вместе с названиями. Даже одно это приравнивает *KStars* к лучшим гидам на экскурсии по ночному небу. Если задержать курсор на второстепенной звезде, то ее название тоже отобразится (если оно есть), или появится астрономическое обозначение, причем используются эффекты постепенного появления и затухания, очень красивые! Щелкнув правой кнопкой мыши на любом объекте, вы получите беспрецедентное количество информации о нем, загруженное прямо из Интернета.



«Я видел сияние C-лучей во тьме вблизи Ворот Тангейзера...»

Благодаря роскошному отображению звезд и их названий, основное окно выглядит как звездная карта, вроде тех, что встречаются в журналах и изданиях для специалистов, и это, вероятно, наивысшая похвала, которой может удостоиться настольный планетарий. Наша галактика представлена в виде многоугольника, протянувшегося через центр карты неба и закрашенного бледно-голубым цветом. Это удобно при поиске звезд, входящих в нее, но не совсем реалистично. Тот же подход используется и для других небесных тел. Галактики и объекты дальнего космоса представлены небольшими зелеными или красными контурами, приближенно показывающими их действительную форму. Огромная туманность Андромеды, например, действительно массивный объект, и ее легко найти без увеличения масштаба, что соответствует реальному объекту ночного неба.

В качестве примера работы с сетью можно рассмотреть загрузку наиболее впечатляющих небесных объектов и отображение их (если возможно) в реальной позиции на карте. Андромеда, подобно многим другим объектам Месье, выглядит как на фотографии с большой выдержкой. Отображение объектов может быть несколько грубым, с неровными зазубренными краями, а чрезмерное приближение к ним приводит к зависанию компьютера. Все потому, что *KStars* использует *KDE/Qt* – вот лучший довод в пользу перехода на *OpenGL* или, возможно, на векторный движок *Arthur* в *Qt 4*.

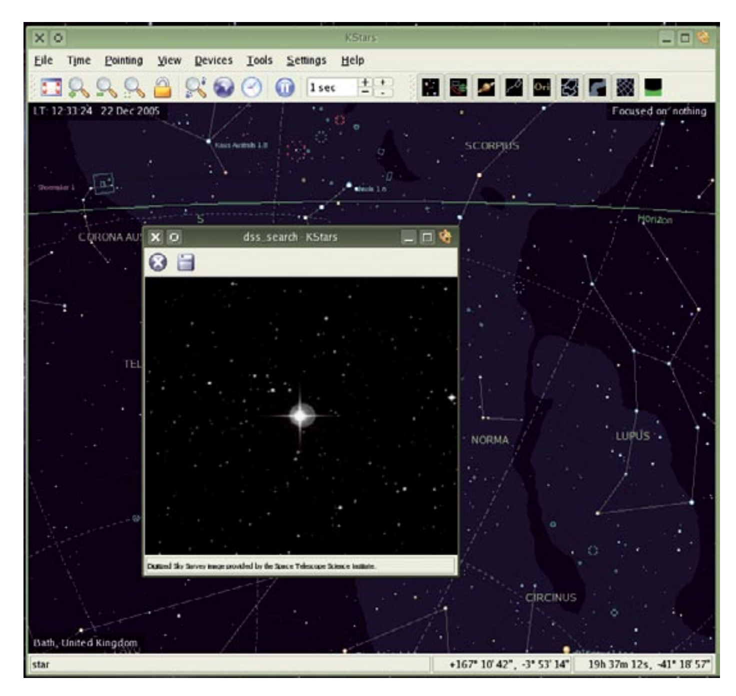
Лунная соната

Луна заслуживает особого разговора, потому что *KStars* имеет определенные проблемы с расчетом фаз Луны, которые не совсем точно накладываются на статическую картинку Луны. Динамика планет вообще практически отсутствует: все наши космические соседи представлены просто картинками, и не сделано даже попытки показать их спутники. Единственным исключением является Юпитер: вы можете открыть отдельное окно, где отслеживается перемещение четырех самых больших его спутников в зависимости от времени. Правда, оно не слишком полезно, если вы хотите точно узнать, что видите в телескопе.

Тот же подход применен для отображения относительного положения планет в Солнечной системе – отдельное окно, в котором показан нисходящий вид планет на орбитах, по которым они вращаются вокруг Солнца, с возможностью переключения между внешней и внутренней группами.

А как насчет оборудования? Что ж, на простейшем уровне есть режим поля обзора, при котором обозначены границы вокруг той области, которую можно увидеть при помощи различных приборов, включая обычный бинокль. Вы также можете управлять телескопом с CCD-матрицей и устройством захвата изображений через протокол INDI: *KStars* сориентирует устройство на объект, наблюдаемый вами в программе, очень даже умно.

Описывая *KStars*, в первую очередь вспомнишь об эффективности, а не о красоте. Он снабжает вас всей необходимой информацией и отлично работает как ценное средство изучения неба. У него огромная база данных звезд и прекрасное отображение. Вы даже можете перевести цветовую палитру в режим для ночного наблюдения – это очень пригодится звездным вечером.



***KStars* может отображать фото с онлайн-сервиса Digital Sky Survey (**Цифровой обзор неба**).**

ВЕРДИКТ LINUX FORMAT

Превосходное решение для астрономов-любителей, вышедших в поле.

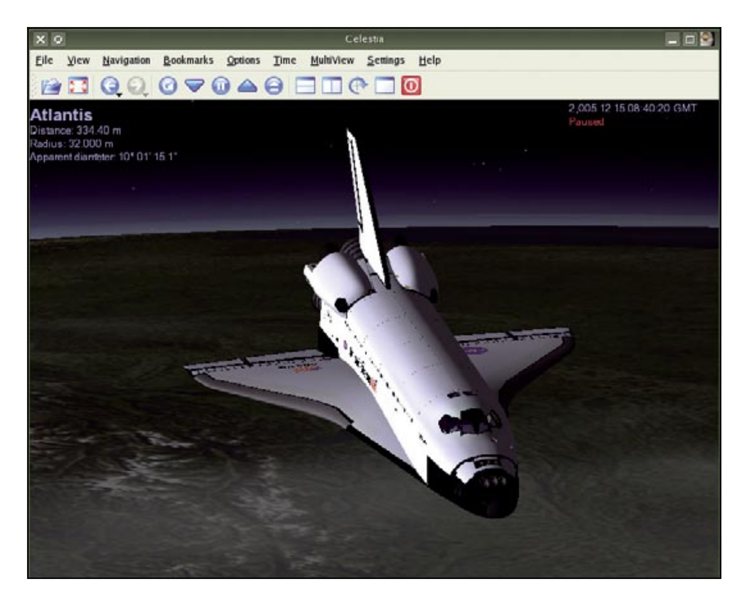
РЕЙТИНГ 8/10

Celestia



Бесконечность – не предел.

• **ВЕРСИЯ:** 1.4.0 • **WEB:** www.shatters.net/celestia
• **Цена:** Бесплатно по лицензии LGPL



Классные дополнения, вроде космического челнока *Atlantis* от SpaceCraft, очень украшают *Celestia*.

Celestia подобна глотку магического звездного зелья, дарящего вам силу супергероя для путешествия сквозь пространство и время. Образно говоря, *Celestia* не приковывает ваши ноги к земле, а сбрасывает кандалы, позволяя вам исследовать любую точку Вселенной и перепархивать с места на место, словно вы обладаете беспредельным невероятным двигателем, но только без невероятности. Пусть вы расцениваете шанс попрыгать через созвездия, подобно капитану «Золотого сердца» (*Heart of Gold*), как излишнюю роскошь – но звезды, видные из вашего сада в этот вечер, *Celestia* тоже покажет хорошо.

Новый исследовательский подход бросается в глаза сразу же после запуска *Celestia*. Интерфейс – в виде стандартного окна KDE, включая панель браузера с закладками; поначалу главное окно сфокусировано на Солнце. Затем оно перемещается на нашу родную планету, которая занимает весь экран, вращаясь в реальном времени. Переход, честно говоря, должен бы быть плавным. Если у вас нет видеокарты с приличным 3D-акселератором, то изображение будет больше похоже на слайд-шоу, нежели на небесный балет. ATI-карты также не свободны от проблем, поскольку разработчики *Celestia* не имели к ним доступа. Но если у вас карта от NVidia, выпущенная в последние два года, то все должно работать прекрасно.

Для отображения *Celestia* использует *OpenGL*, и производит рендеринг в зависимости от характеристик вашей машины, включая последние возможности *OpenGL 2.0*.

Итак, сперва вы смотрите из космоса на Землю с расстояния примерно 32 000 км. Рендеринг объектов иначе как «прекрасным» не назовешь; текстурирование безупречно, вплоть до того, что на поверхности океанов виден блик от Солнца. Вдобавок текстуры можно менять, причем процветающее онлайн-сообщество *Celestia* создает их на базе изображений репозитария NASA, благодаря которым Земля показана невероятно детально. *Celestia* способна переключаться между уровнями детализации динамически, используя слабо детализированное изображение, например, юго-запада Англии и изображение с высоким разрешением, если вам нужно разглядеть полуостров Лизард в Корнуолле.

Выбор в меню Navigation (Навигация) пункта Go To Surface (На поверхность) перенесет вас на *terra firma*, с традиционным для планетариев видом небесной сферы. Вы справедливо подумаете, что имея всю Вселенную в своем распоряжении, рассматривать ее с Земли – отсталая идея. Конечно, спуск на поверхность подразумевает поверхность не только Земли, можно причалить и к другим планетам, например, к Марсу, где наблюдается прекрасный оранжевый восход.

Вы свободны в выборе широты и долготы, как в любом настольном планетарии, но все-таки наземный вид – явно не главный повод выбрать *Celestia*.

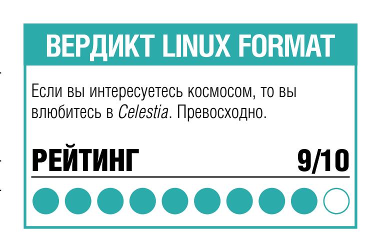
Человек-ракета

Самое лучшее применение программы – это полеты в пространстве. Положение каждой звезды, планеты или астероида, содержащихся в базе данных *Celestia*, задано тремя координатами, то есть просто щелкните в том месте, куда собрались отправиться (хоть в другую галактику) – и всего за несколько секунд вы пронесетесь сквозь бездны космоса и окажетесь на орбите вблизи выбранного объекта. Там можно развернуться и взглянуть на место, откуда вы прибыли – совершенно уникальная перспектива созерцать чужое звездное небо. Включив отображение линий, соединяющих звезды в созвездиях, можно даже наблюдать, как они искажаются по мере удаления от Земли. Ручное управление позволяет ускорять или замедлять движение и передвигаться так, словно вы находитесь в космическом корабле.

Интерфейс пользователя почти идеален. Главное окно поддается разбиению на сколько угодно частей, причем одна из хитростей – разбиение окна на две горизонтальные панели и небольшой сдвиг одной из них для получения объемного вида [к сожалению, мне не удалось этого добиться, -прим. перев.]. Можно пометить закладкой любое местоположение, как в web-браузере, и поделиться ею с другими пользователями *Celestia*. Все экраны можно синхронизировать, а также сфокусировать на одну цель. Единственным упущением является отсутствие поиска, хотя имеется прекрасный небесный браузер.

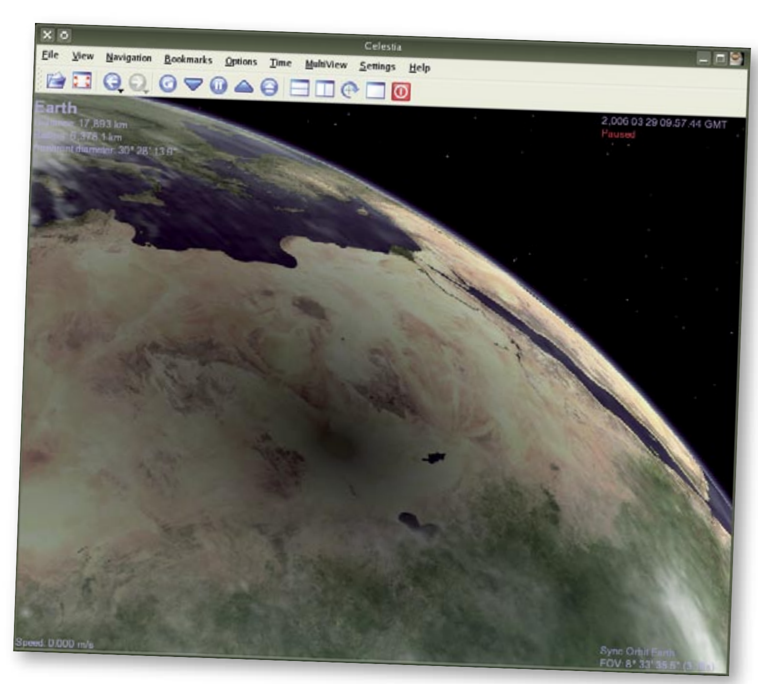
Эти возможности – только вершина айсберга. Далекие галактики составлены из частиц и выглядят так, как и должны быть видны с Земли, но когда к ним приближаешься, они распадаются на пылевые рукава, в соответствии с углом зрения. Учтены даже эффекты, связанные с конечной скоростью света. *Celestia* использует абсолютно оригинальный подход к настольному планетарию, и в результате оно предоставляет уникальный взгляд на связи между различными небесными телами и на то, как они взаимодействуют друг с другом. И самое важное – оно пробуждает страсть к исследованию.

«ВЫ ПРОНЕСЕТЕСЬ СКВОЗЬ БЕЗДНЫ КОСМОСА ЗА СЧИТАННЫЕ СЕКУНДЫ.»


ВЕРДИКТ LINUX FORMAT

Если вы интересуетесь космосом, то вы влюбитесь в *Celestia*. Превосходно.

РЕЙТИНГ 9/10



Можно увидать сказочное зрелище – например, затмение в Сахаре.

Xephem

Научная школа старых мастеров.

• **ВЕРСИЯ:** 3.7.1 • **WEB:** www.clearskyinstitute.com/xephem
• **Цена:** Бесплатно для персонального использования

***Xephem* (произносится как «икс-и-** фем») – дедушка астрономических программ в Linux. Его корни в лабораториях UNIX – обязательно с громоздкой дубовой мебелью, где у системных администраторов свободного времени в избытке, и они разрабатывают действующую солнечную систему для окна About (О программе) просто потому, что умеют это делать.

Интерфейс пользователя напоминает уровень 1990-х, но само приложение не остановилось в развитии. Оно больше похоже на коробку с секретом, чем на комплексный настольный планетарий типа *KStars*. Первое окно, которое вы видите, показывает время – и очень точное. Это локальное время, использующее ваши координаты на планете, затем Юлианское, всемирное и сидерическое звездное время, все они очень важны для астрономических лабораторий. Представлена и менее возвышенная информация, например, местное время восхода и захода Солнца, что делает *Xephem* лучшей утилитой в нашем обзоре по части быстрого получения специфических данных.

Научный подход сохранен и в прекрасных видах Луны и планет, которые открываются в отдельном окне, отображающем и фазу объекта, и интересные элементы его поверхности. *Xephem* делает это лучше всех прочих – графику вряд ли можно назвать реалистичной, зато она облегчает идентификацию элементов, не распознанных вами в телескоп.

Отображение местности также весьма жизненно и включает любопытные детали, например, место посадки *Beagle* 2 на Марс. Кстати, *Xephem* является одним из лучших инструментов для изучения видимых объектов на Красной планете, так как предоставляет полный список кратеров и морей, которые четко помечены и легко различимы. Если у вас ничего нет, кроме бинокля, то имеет смысл начинать учебу здесь. Вид Night At A Glance (Взгляд в Ночь) также выполнен безупречно, у всех основных планет отображается граница день/ночь в зависимости от времени.



***Xephem* буквально набит функциями.**

Дотошные лабораторные крысы снабдили *Xephem* продвинутыми возможностями, вроде решения уравнений или online-анализа кривых светимости, которые придают *Xephem* строгий академический вид: вы даже можете управлять некоторыми моделями телескопов.

Но и для начинающих это превосходное приложение. Тут нет графических изысков *Kstars* или *Celestia*, зато имеется действительно отточенный лаконичный GUI, предоставляющий всю необходимую информацию в понятной форме.

ВЕРДИКТ LINUX FORMAT

Не дайте старомодному интерфейсу оттолкнуть вас, приложение просто отличное.

РЕЙТИНГ 7/10



Maestro

Практический инструмент для исследования Марса.

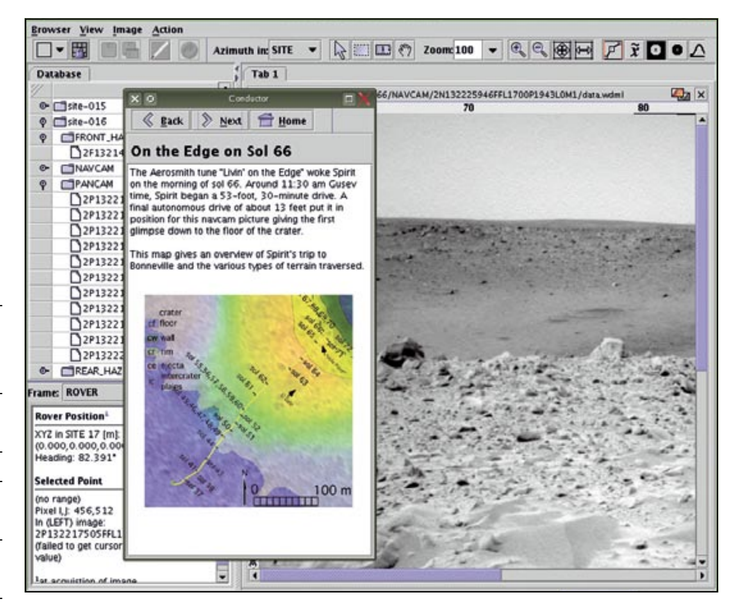
• **ВЕРСИЯ:** 2004.01 • **WEB:** http://marsrovers.jpl.nasa.gov
• **Цена:** Бесплатно для персонального использования

Maestro – Java-приложение, разрабо- танное в лаборатории реактивных двигателей для нужд последней миссии NASA на Марсе – Mars Exploration Rover. Фактически это часть программного обеспечения, разработанного для управления двумя марсоходами, изучающими и картографирующими поверхность Марса, и некоторое время данная версия подкачивалась потоком информации прямо с них. Ученые используют его для составления плана дальнейших передвижений, а мы с вами обследуем с его помощью поверхность планеты и изучим данные, получаемые от обоих роботов.

Благодаря чрезвычайному успеху и долголетию марсианской миссии, *Maestro*, несмотря на прекращение финансирования, все еще предоставляет доступ к этому удивительному проекту.

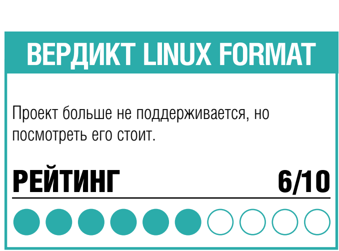
При запуске в приложении нет данных, их придется скачать отдельно. Это касается даже тестового набора данных, использовавшегося марсоходами до начала миссии. После тренировки на тестовом наборе вы поймете, чего ожидать от реальных данных. Попривыкнув к интерфейсу, просто загрузите данные интересующей вас миссии – обычно в виде пакета нескольких марсианских суток (кстати, их продолжительность 24 часа 39.5 минут) – и выбирайте любые на боковой панели.

Как только данные загрузятся, наметьте путь каждому из марсоходов – и будете получать реальные изображения от их бортовых камер. Подробная информация о характере поверхности и задачах каждой миссии прилагается, и между этими данными можно свободно переключаться.



***Maestro* позволяет интерпретировать данные, прибывшие прямо с Марса.**

Наиболее впечатляющий аспект программы – возможность наложить изображения на приблизительную трехмерную сетку местности около каждого марсохода, рассчитанную при помощи его сенсоров. Даже жутковато наблюдать столь пронзительно-реальный инопланетный ландшафт, комфортно расположившись за Linux-компьютером. Восприятие изображений, столь знакомых благодаря печатным изданиям последних лет, обретает новизну. *Maestro* предоставляет сырые, необработанные данные, и ваше воображение может додумать все что угодно – от мелких песчинок до маленьких зеленых человечков, прячущихся за скалами.



ВЕРДИКТ LINUX FORMAT

Проект больше не поддерживается, но посмотреть его стоит.

РЕЙТИНГ 6/10

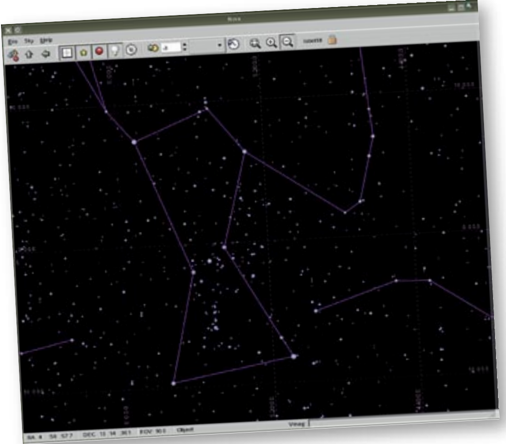
Nova



НА ДИСКЕ

Приложение Gnome нового поколения.

• **ВЕРСИЯ:** 0.1.0 • **WEB:** http://nova.sourceforge.net • **Цена:** Бесплатно по лицензии GPL



Дизайн прост, но программа не простецкая.

Несмотря на то, что приложение названо в честь катаклизма, знаменующего конец белого карлика [взрыв сверхновой или supernova в англоязычной терминологии, -прим. ред.], это заурядный вариант настольного планетария. Вы получаете вид на небо с Земли и минимальный интерфейс пользователя. К неудовольствию звездочетов, с установкой могут возникнуть проблемы, так как проект разбит на три отдельные библиотеки. Необходимо соблюдать порядок сборки, плюс каждая имеет свои собственные зависимости. Придется также запустить скрипт, чтобы загрузить последние базы данных. Это отличный способ поддерживать их в актуальном состоянии, но довольно нудное занятие, если хочется просто поработать с приложением.

Тем не менее приложение-планетарий вполне достойное. Интерфейс пользователя типичен для многих современных Gnome-приложений (т. е. минималистичен) и обладает большим потенциалом, если к проекту подключить свежие силы. Точки, отображаемые на небе, увеличены относительно пропорциональных размеров объектов, и звезды и планеты легко отыскать – весьма удобно при использовании приложения в полевых условиях на ноутбуке. Вы можете управлять своим телескопом и устройством слежения, а также задавать расписание наблюдений, чего нет ни в одном из рассматриваемых в нашем обзоре приложений.

ВЕРДИКТ LINUX FORMAT

Первые шаги, но в будущем обещает многое.

РЕЙТИНГ 5/10





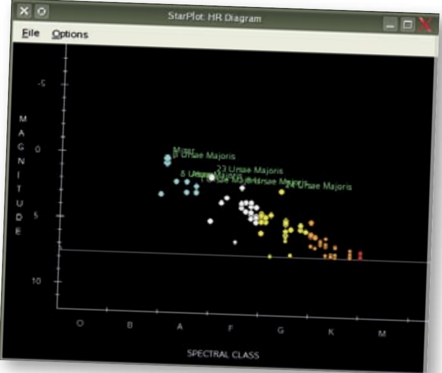

StarPlot

НА ДИСКЕ

Звезды на палочках.

• **ВЕРСИЯ:** 0.95.3 • **WEB:** http://starplot.org • **Цена:** Бесплатно по лицензии GPL

Можно ограничить число звезд на экране по их величине.



Если вам случалось играть в игру Дэвида Бребена (David Braben) о космической торговле Frontier, или в ее предшественницу Elite (а если нет – ступайте к Майку Сондерсу на правеж), то вы знакомы с подходом к отображению звездной системы, применяемым в *StarPlot*. Он аналогичен «местному радарному обзору» (local radar view) и навигации между системами в этих играх.

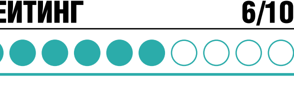
Звезды помещаются на сфере, рассеченной плоскостью на уровне середины вертикальной оси. Высота звезды над или под этой плоскостью обозначена перпендикуляром, соединяющим звезду и плоскость, и легко разобраться со взаимным расположением звезд. Причем угол зрения можно менять – например, установить его таким образом, чтобы смотреть прямо сквозь плоскость Галактики: тогда все звезды будут располагаться выше или ниже центральной точки. Щелкнув на подписи к звезде, можно отфильтровать в реальном времени все звезды данного типа. При этом в отдельном окне отображаются величины звезд для каждого спектрального класса.

StarPlot – прекрасное средство для просмотра распределения звезд различных классов и их расположения друг относительно друга. Благодаря расширяемой базе данных звезд и продуманному интерфейсу пользователя это прекрасный экземпляр двумерного настольного планетария.

ВЕРДИКТ LINUX FORMAT

Функций мало, но те, что есть, работают хорошо.

РЕЙТИНГ 6/10

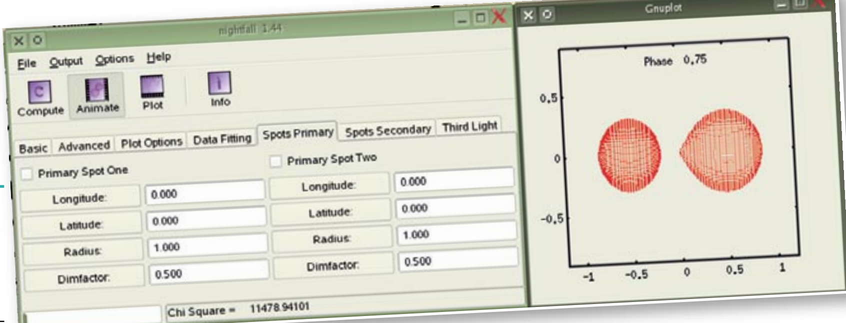


Nightfall

НА ДИСКЕ

Модельер командной строки.

• **ВЕРСИЯ:** 1.44 • **WEB:** www.lsw.uni-heidelberg.de/users/rwichman/Nightfall.html
• **Цена:** Бесплатно по лицензии GPL



О, Азимов! Даже дилетанты научились измерять интервал между двойными звездами.

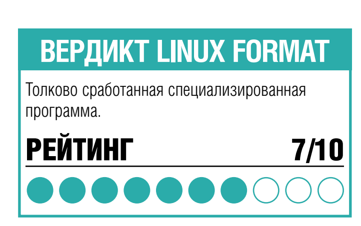
Если, по-вашему, экономный интерфейс Xephem – и тот слишком перегружен, то вот вам программа: при запуске *Nightfall* из командной строки даже окно не открывается. *Nightfall* – прежде всего научное приложение, предназначенное для анимации затмений в системах двойных звезд. Звезды таких систем расположены столь близко, что они вращаются по орбитам вокруг общего центра масс (называемого барицентром). Иногда получается, что одна звезда заслоняет другую, в итоге на кривой светимости появляется провал, причем через равные промежутки времени. *Nightfall* умеет обрабатывать кривые светимости и на основании затмений строить модель удаленной двойной звездной системы.
Кривые светимости прилагаются как файлы с данными к главному пакету, а интерфейс пользователя позволяет изменять данные системы. Вы можете анимировать результаты и увидеть две звезды, вращающиеся вокруг барицентра, или получить поперечное сечение в любой точке двойной системы.

Даже если у вас нет специальных знаний, но есть хотя бы мимолетный интерес, программа предоставит достаточно информации, чтобы помочь вам разобраться в этих удивительных системах.

ВЕРДИКТ LINUX FORMAT

Толково сработанная специализированная программа.

РЕЙТИНГ 7/10





АСТРОНОМИЧЕСКИЕ ПРОГРАММЫ ВЕРДИКТ

Полетав среди звезд, мы оставили два приложения, претендующих на победу: *KStars* и *Celestia*. За *KStars* будут те, кому нравится наблюдать небо на природе: они получают ясное схематическое изображение неба, большую базу данных звезд и идеальное планирование ночных наблюдений. Дополнительные возможности (управление телескопом и устройством слежения) вкупе с ночной цветовой палитрой делают *KStars* наилучшим Linux-инструментом астронома-любителя.

Celestia принадлежит к другому классу программ. Это не настольный планетарий. Заставлять его показывать созвездия, видимые с заднего двора, попросту глупо. *Celestia* – удивительное приложение, переполняющее вас ощущением чуда, которое испытывает ребенок, глядя на ночное небо. Вы можете отправиться в путешествие с Земли, облететь наших небесных соседей, увидеть, как их заливает свет далекого Солнца, а затем перекочевать в ближайшее скопление звезд и даже в другие галактики. Приложение создает реальнейшее чувство пространства, и вы уже ни за что не вернетесь к старому двумерному представлению неба.

Вот почему *Celestia* – наш победитель. Возможно, это не самый практичный инструмент из имеющихся в наличии, но зато он больше других способен разбудить интерес будущих и настоящих астрономов за рабочим столом, и послужит чудесным средством обучения.

Проще говоря, *Celestia* вне этого мира.

ВАШ ВЫБОР

Расскажите о приложениях, которые оказались полезны вам, написав в английскую редакцию lxf. letters@ futurenet. co. uk

CELESTIA 9/10

Это клише, но *Celestia* достигает высот, недоступных любому другому астрономическому приложению.

ИЩЕТЕ ЛУНУ В ПОДАРОК ЛЮБИМОЙ? ВЫБИРАЙТЕ!

Фото Луны и три картинки из приложений сделаны одновременно. Результат впечатляет…



Celestia



KStars



Xephem



Реальная фотография

СРАВНИТЕЛЬНАЯ ТАБЛИЦА

	Celestia	KStars	Maestro	Nightfall	Nova	StarPlot	Xephem
Примечания к		✔					✔
Управление CCD-устройствами		✔			✔		✔
Реалистичное отображение	✔	✔	✔	✔			
Загрузка из сети	✔	✔	✔	✔		✔	
Вид созвездий	✔	✔			✔	✔	✔
3D-пространство	✔		✔	✔			
Поддержка скриптов	✔	✔				✔	
Поиск	✔	✔	✔				✔
OpenGL	✔		✔	✔			
Фазы Луны	✔	✔					✔
Лунные вычисления	✔						✔







Майк Сондерс (Mike Saunders) В этом месяце Майк попивал шоколадно-вишнёвый Моцарт-коктейль (да, есть такой!).

Сотни программ с открытым кодом создаются и обновляются каждый месяц, поэтому среди них непросто найти достойную использования. В разделе **HotPicks** мы просеиваем все последние релизы и отделяем зёрна от плевел, отбирая для обзора только самое лучшее. И ежемесячно самое впечатляющее приложение выигрывает наш приз: Hottest Pick.

Вы можете сами попробовать все программы, вошедшие в наш обзор, установив их с приложенного диска. Если вы видели другие отличные программы, которые стоит включить в обзор HotPicks, черкните мне письмо на **mike.saunders@futurenet.co.uk.**

HOTPICKS В ЭТОМ РАЗДЕЛЕ

ОБРАТИТЕ ВНИМАНИЕ НА ПОБЕДИТЕЛЯ HOTPICKS

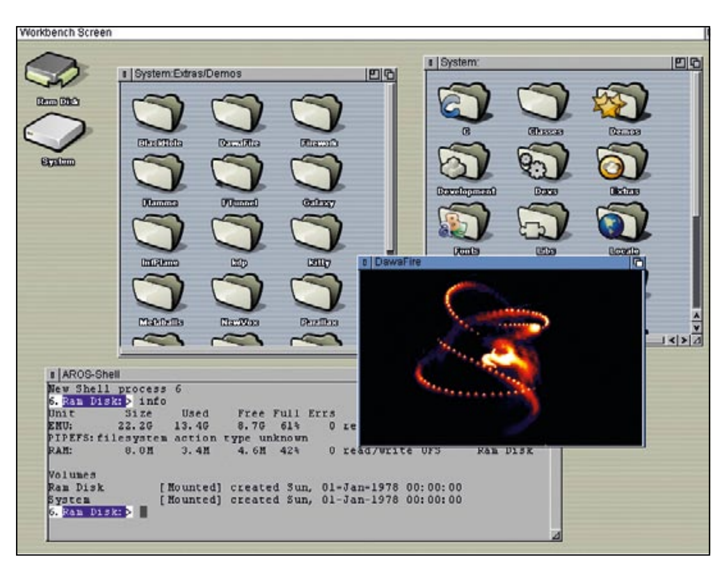
В нашем разделе **HotPicks**… все заслуживает внимания, но ежемесячно мы награждаем один проект за его выдающиеся свойства. Выбирается ну самый лучший!



ОПЕРАЦИОННАЯ СИСТЕМА

Aros

• **ВЕРСИЯ** 2005-12-07 • **WEB** www.aros.org



Aros быстр и отзывчив, как и полагается системе на базе AmigaOS. Включайтесь в список рассылки, если есть желание помочь.

После кончины Commodore International многие фанаты Amiga занялись поиском подходящей для их сообщества системы, которая была бы вне контроля тогдашних компьютерных гигантов (то есть Microsoft и Apple, как и сейчас). В результате немало экс-пользователей Amiga (изрядная часть команды LXF) переключились на Linux. Однако небольшая команда кодеров не сдалась: в середине девяностых Аарон Дигулла (Aaron Digulla) и группа его разработчиков начали работу над Aros – клоном AmigaOS с открытым исходным кодом.

Aros (Amiga Research Operating System) намерен стать полноценным клоном AmigaOS, совместимым с AmigaOS 3.1. Одна из целей проекта – межплатформная переносимость, так что в конечном счёте Aros будет работать на обычных PC, Mac и прочих. Когда команда Aros приступала к разработке, для PC почти не было ОС с открытым кодом – только ранние релизы Linux- и BSD-систем, поэтому они решили реализовать свой клон поверх Linux и X. По сути, Aros работал как обычное X-приложение, да и сейчас им остается. Однако действительно впечатляющий результат – включение собственной операционной системы.

> «ЭТОТ РАБОЧИЙ СТОЛ ВЕРНЕТ ВАС В 1985-Й ГОД.»

Содрать драйверы

Да, Aros будет работать на голом PC, загружаясь с Live CD. По сравнению с Linux и FreeBSD набор драйверов невелик (графика VESA и поддержка наиболее распространённого оборудования; но ситуация улучшается). Система работает на большинстве ПК. Чтобы посмотреть её в деле, запишите на CD ISO-образ системы, взяв его с нашего диска, и загрузите ваш компьютер с этого Live CD. Если он не заработает, вы всегда можете распаковать архив **linux-system.tar.bz2** и запустить **Aros** из-под Linux.

Ретростальгия

Если вы поклонник Amiga, то созерцание загрузки Aros в быстрый и знакомый, выполненный в стиле *Workbench*, рабочий стол вернет вас в 1985-й год. Aros использует открытую замену *Workbench*, названную *Wanderer*. Щелчок правой кнопкой мыши вызовет меню и пару иконок на рабочем столе.

Навигация в системе и управление окнами в точности как у AmigaOS – клонировать так клонировать. MUI (Magic User Interface – магический пользовательский интерфейс) заменен на похожий инструментарий *Zune*; MUI давным-давно используется в приложениях AmigaOS, так что иметь его эквивалент в Aros очень удобно. Файловая система Aros и оболочка до мелочей повторяют AmigaOS: броские иконки создают иллюзию, что вы сидите за A1200.

В систему включены разнообразные мелкие приложения и демо-программы: графические заставки, утилиты для рабочего стола и всякая всячина из Aminet (архив программ для Amiga). Пока что программы приходится перекомпилировать для Aros x86, но одна из целей проекта – обеспечить бинарную совместимость, и Aros будет полностью портироваться на 68k Amiga. В общем, он вполне Amiga-подобный, а если дать ему поразвиваться ещё пару лет – станет маленькой, но милой независимой ОС для рабочего стола.

Созрел ли Aros для повседневной работы с экс-пользователями Amiga? Вряд ли; надо по крайней мере обеспечить такие важные компоненты, как полноценная поддержка сети и набор собственных инструментов для разработчика (оба уже на подходе). Но в любом случае этот проект – изумительное достижение небольшой команды, и изучать его – одно удовольствие. Возможно, Aros – наилучший способ продлить жизнь Amiga: объединить скорость и простоту AmigaOS с непобедимым превосходством в соотношении цены и производительности для PC.

ОРГАНАЙЗЕР

TreeLine

• **ВЕРСИЯ** 0.14.0 • **WEB** www.bellz.org/treeline

Мы, люди, забывчивые создания. При хроническом дефиците времени в наши дни сложно упомнить квадрильоны контактов, списков, учетных записей и пунктов к выполнению, которые жужжат у нас в голове. Программы PIM (персональная информационная система) помогают в этом деле, но они склонны ограничиваться конкретным типом данных: деловые встречи, почта и т.д. В *TreeLine* применён другой подход, позволяющий хранить любой набор данных в древовидной структуре. Все, что вам хочется организовать, можно ввести и экспортировать во всевозможные форматы. Программа написана на Python с использованием библиотеки Qt для пользовательского интерфейса, так что вам понадобится установить *PyQt*.

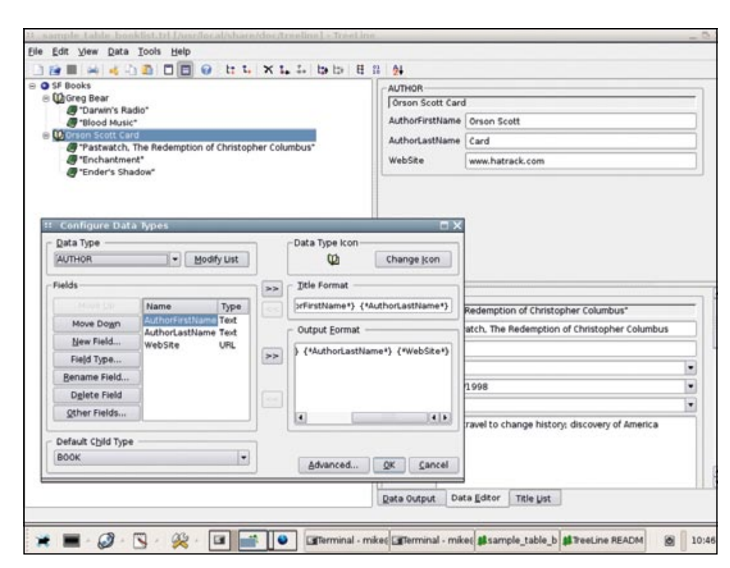
При первом запуске *TreeLine* отобразит панель, содержащую дерево элементов, их содержимое и вкладку для переключения между режимами редактирования и вывода данных. Панель инструментов с подсказками слегка оживляет интерфейс, но она слишком мелкая и не вполне очевидна для новичков. К счастью, существует онлайновый файл README: в нем с помощью экранных снимков иллюстрируется навигация по интерфейсу. Ещё более полезен набор файлов с примерами создания коллекций информации.

В основу *TreeLine* положены узлы и поля, вместе они формируют мини-базы данных, способные хранить информацию различных типов (обычный текст, HTML, логические типы и т.п.). Поля могут также ссылаться на другие узлы. Гибкий диалог Data Types позволит вам создать и организовать поля, и определить способ отображения поля в окне вывода данных Data Output. Подобный уровень гибкости позволяет хранить почти любой набор или список элементов.

Пожалуй, наиболее полезная возможность *TreeLine* – коллекция фильтров экспорта, записывающих данные в HTML, XML или разделённый табуляцией текстовый формат (родной формат *TreeLine* – диалект XML). Просмотрите файл readme.trl (созданный в редакторе приложения), пример применения способностей *TreeLine*.

TreeLine – нечто среднее между PIM и базой данных: он гораздо более гибок, чем программы первого рода, и при этом проще программ второго рода. Достоин изучения, если вам нужно структурировать разнородную информацию в одном месте.

«TREELINE – ЗОЛОТАЯ СЕРЕДИНА МЕЖДУ PIM И БАЗОЙ ДАННЫХ.»



Создание вашей коллекции с нуля через интерфейс *TreeLine* может озадачить, так что просмотрите файлы с примерами.

МЕНЕДЖЕР РЕЦЕПТОВ

Krecipes

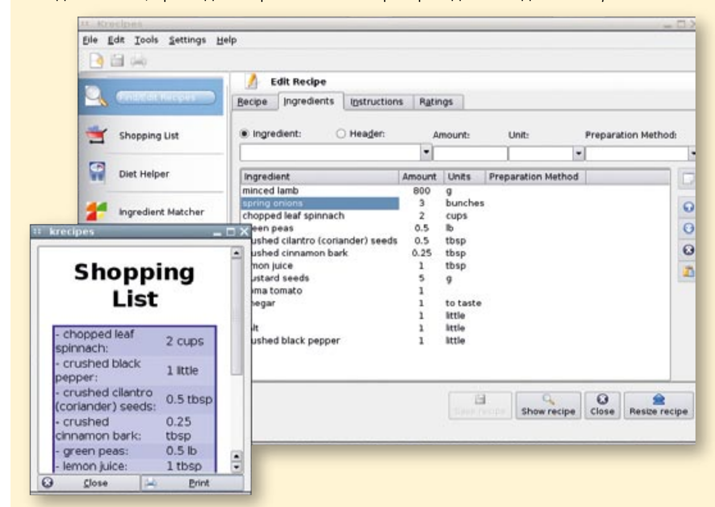
• **ВЕРСИЯ** 09 • **WEB** http://krecipes.sourceforge.net

Гэри Родес (Gary Rhodes), Кейт Флойд (Keith Floyd), Hamburgler – известные шеф-повара, способные, нам на радость, превращать пошлые продукты в деликатесы, при виде которых слюнки текут. Однако мы, компьютерщики, зациклились на программировании, а искусство приготовления пищи кажется нам недоступным. К счастью, компьютеры пригодятся и здесь. *Krecipes* поможет вам спланировать своё питание, проверить ингредиенты и составить список покупок. В вашем буфете остались только соевый соус, рыбные консервы и крекеры? Не расстраивайтесь, *Krecipes* придумает за вас, как состряпать из них приличное блюдо.

По имени Krecipes вы, конечно, угадали, что он использует KDE, поэтому компиляция из исходных текстов потребует установки пакетов разработчика. Для хранения данных Krecipes можете использовать базу данных *MySQL* (а если не можете, в ход пойдет встроенный SQLite). Мастер запуска проведёт вас через шаги выбора базы данных, четыре примера рецептов и описание, как скачать данные о четырёхстах ингредиентах – симпатичный гид!

Главное окно *Krecipes* – хорошо отделанное и веселое, богатое красками; оно снабжено также деревом для навигации среди блюд. Рецепты отсортированы по категориям: паста, супы и т.д. Встроенных рецептов в *Krecipes* только четыре; вам придётся создать свою базу рецептов, вводя их в экране Edit Recipes, где панель со вкладками и выпадающие списки помогут моментально создать или поменять их. Приложение поддерживает импорт рецептов в форматах MasterCook, MealMaster и RecipeML. Рецепты отображаются красиво, ингредиенты и информация о калорийности выносятся в отдельную рамку. *Krecipes* отлично подходит для составления диетического меню – вы выбираете несколько блюд, а он готовит к распечатке список необходимых продуктов. Или – вы устанавливаете себе дневную норму витаминов, а *Krecipes* генерирует список подходящих блюд. Расширенный интерфейс позволит вам выбрать граммы или унции, свойства блюд и авторов рецептов, вы также сможете упорядочивать рецепты по вкусности и вставлять комментарии, например, «Отлично идет с собачьим кормом!».

Krecipes – одно из самых законченных приложений, попавшихся нам за последние несколько месяцев. Он явно был создан программистами – любителям поесть: набор возможностей и толковые распечатки делают его воистину полезным приложением для ежедневного использования. Советуем попробовать эту программу шеф-поварам, использующим Linux – только не держите свой ноутбук слишком близко к плите, не то произойдет кулинарное бедствие...



Борща и блинов нет, но это можно исправить.

ПАКЕТ ВЕКТОРНОЙ АНИМАЦИИ

Synfig

HOTTEST PICK!

• **ВЕРСИЯ** 0.61.03 • **WEB** www.synfig.com

Одним из наиболее полезных преимуществ приложений с открытым кодом является возможность вдохнуть новую жизнь в заброшенные проекты. Большинство проприетарных программ чахнут и впадают в безвестность, если авторы ими больше не занимаются. Однако открытие кода дает проекту шанс воскреснуть.

Именно так произошло с *Synfig* – набором приложений для анимации векторной графики. Из-за финансовых трудностей программист *Synfig* Роберт Квоттлбаум (Robert Quattlebaum) более не мог работать над этим проектом с полной занятостью; он отдал свой код сообществу и, таким образом, способствовал возрождению проекта.

Synfig предназначен для создания анимации кинематографического качества. Он способен генерировать мягкие переходы между узловыми кадрами, избавляя художника-мультипликатора от необходимости индивидуально прорисовывать каждый кадр. Квоттлбаум находился под значительным влиянием *аниме*, но программа столь же полезна и для традиционных мультиков. Две ее ключевых возможности – это гордо именуемые spatial resolution independence (независимость от пространственного разрешения), позволяющая отрисовывать кадры с достаточной резкостью при любом размере, и temporal resolution independence (независимость от разрешения во времени), создающая мягкие переходы между кадрами при любой их частоте.

Установка

Synfig поставляется в двух частях: утилита командной строки *Synfig*, позволяющая обрабатывать родные XML-файлы без графического интерфейса, и *Synfig Studio*, предоставляющая этот интерфейс на базе GTK. Поскольку *Synfig* написан на C++, вам понадобятся libxml++ и libsigc++, а также C++ GUI- библиотеки для Studio (glibmm, gtkmm и т.п.). Архивы не включают никакой документации по установке, однако стандартной процедуры *configure*, *make* и *make install* (под root) вполне достаточно. После этого вы сможете запустить приложение, набрав команду *synfigstudio*.

«СООБЩЕСТВО OPEN SOURCE ПОЛУЧИЛО ЦЕННЫЙ ПОДАРОК.»

С первого взгляда интерфейс *Synfig* сильно напоминает *Gimp*: небольшое окно с инструментами дополняется рядом других «плавающих» окон. Сторонников у такого GUI не слишком много, и, к сожалению, в нём нет присоединяемых (dockable) окон, вошедших в обиход с *Gimp 2*. Поэтому для получения максимума отдачи от интерфейса *Synfig*, чтобы не блуждать по перекрывающимся окнам, вам понадобится разрешение по крайней мере 1024x768 (и желательно неброский оконный менеджер).

Инструменты Navigator, по умолчанию размещаемые в правой части экрана, обеспечивают быстрый доступ к различным фигурам анимации (которые вы можете назвать и сами), а также к браузеру палитр и списку предыдущих действий (*Synfig* поддерживает многоуровневый откат).

РИСОВАНИЕ В SYNFIG



Как и многие другие подобные пакеты, *Synfig* использует слои: в *Synfig* каждая фигура (shape), которую вы создадите, является слоем. Инструменты позволяют создавать векторные многоугольники, круги и кривые Безье, можно также рисовать от руки (*Synfig* сам определит точки, которые позволят потом подобрать форму). Для раскраски предоставляются стандартные инструменты заполнения и градиенты. В состав программы включены несколько фильтров, например, motion blur, и bevel, а также различные искажения вроде растягивания и масштабирования.

При создании новой анимации можно указать продолжительность (в секундах) и частоту кадров, а затем переключиться в режим пред-просмотра, чтобы увидеть уменьшенный в размере результат. Быстрота просмотра зависит от вашего процессора и графических драйверов — для сложной анимации лучше использовать процессор с тактовой частотой не ниже 1.5 ГГц. В режиме редактирования можно переключаться в любой кадр с помощью ползунка временной шкалы внизу. Закончив работу, вы сможете сохранить индивидуальные кадры или всю анимацию как серию файлов в форматах BMP, JPEG, GIF и PNG (помимо прочих). Вы также можете создать MPEG-видеоролик с помощью библиотеки *ffmpeg*.

Забираясь внутрь

К сожалению, на данный момент не существует онлайн-документации, хотя имеется руководство в формате PDF, объясняющее базовые принципы использования программы. В большинстве случаев, чтобы разобраться с какой-либо продвинутой возможностью *Synfig*, потребуются эксперименты. Множество диалогов и опций помечены как нереализованные, что отражает текущий статус – версия 0.6x.

Однако мы не считаем, что вам понадобится много подсказок, чтобы понять, что могут делать инструменты *Synfig*. *Synfig* – удивительное достижение; сообщество Open Source получило ценный подарок. Беспокойство вызывает только стабильность программы, поскольку она имеет тенденцию зависать чаще, чем хотелось бы. Здесь может помочь только регулярное сохранение вашей работы. В целом же, если отвлечься от зависаний, программа ведёт себя вполне прилично. Важнее всего тот факт, что разработка *Synfig* ориентировалась на нужды художников, и хотя интерфейс слегка злоупотребляет компьютерным жаргоном, большинство пользователей сочтет его простым и достаточно удобным. Отличная работа.



***Synfig* поставляется с отличными примерами (вроде этих пиратов), демонстрирующими многосторонность программы..**

HOT GAMES

ГОНКИ

Wrong Way

• **ВЕРСИЯ** 0.2.6 • **WEB** http://kvik.sh.cvut.cz/~opi/python

Вы, наверное, полагаете, что в наши дни изобилующих полигонами трёхмерных игр и неуклонно улучшающихся графических карт уже нет места для гонщика старой школы. Но даже банальные двумерные гонки могут стать отличным времяпрепровождением – возьмите, к примеру, *Micro Machines* в режиме нескольких игроков. *Wrong Way* (Дорога не туда) предоставляет более реалистичные машины и настройки, чем MM и Ко, и включает несколько знаменитых машин, которым вы сможете спалить покрышки. Для установки игры понадобится *Python* и привязки *PyGame*. Распаковав архив, запустите **./run.py** в каталоге **wrw** – и вы окажетесь в главном меню. Экран содержит логотипы, созданные скриптами *Gimp*; музыка довольно скверная, а очертания автомобилей условны – графика явно не самая сильная сторона игры. Кое-какие настройки позволяют менять разрешение и переопределять кнопки управления.

Wrong Way включает режимы одного и двух игроков. В однопользовательском режиме на данный момент можно только соревноваться на время, а в режиме двух игроков вам предоставляется разделённый на две части экран, позволяющий посостязаться с другом. К сожалению, в игре не предусмотрены компьютерные игроки.

В игру включены три трека: извилистая трасса с большим количеством ограждений, об которые можно ободрать бока, и брошенных на обочинах автомобилей; трасса-восьмерка, обвивающая деревья и болота; чертовски трудный овал с двумя сужениями размером не шире машины. Виды вокруг трека вполне приличные – деревья приемлемы, изображения зданий встречаются вполне убедительные, но самое важное – они не отвлекают от игры. Вы можете выбирать из шести машин, включая элегантный Bentley и шикарную Mazda RX-7.



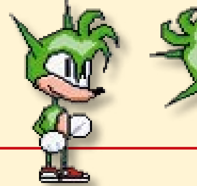
Несмотря на свое имя, игра не карает за выбор неверного пути (водитель красной машины, вы слышите?). Обман трудящихся...

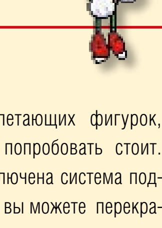
Управление в *Wrong Way* представляет собой тщательно настроенный баланс быстрого маневра на высокой скорости и сброса скорости на поворотах, при правильном их прохождении. Столкновения, однако, несколько пародийны – вы просто тупо останавливаетесь.

Игра чуткая, требует собранности, а в режиме двух игроков представляет собой отличное развлечение, причем на своей половинке экрана нужна молниеносная реакция: повороты возникают буквально из ниоткуда. Добавить в нее хотя бы базовых компьютерных игроков – получатся отличные гонки.

МОДЕЛИРОВАНИЕ ПОПЕРЕЧНОГО ДВИЖЕНИЯ

SDLjump



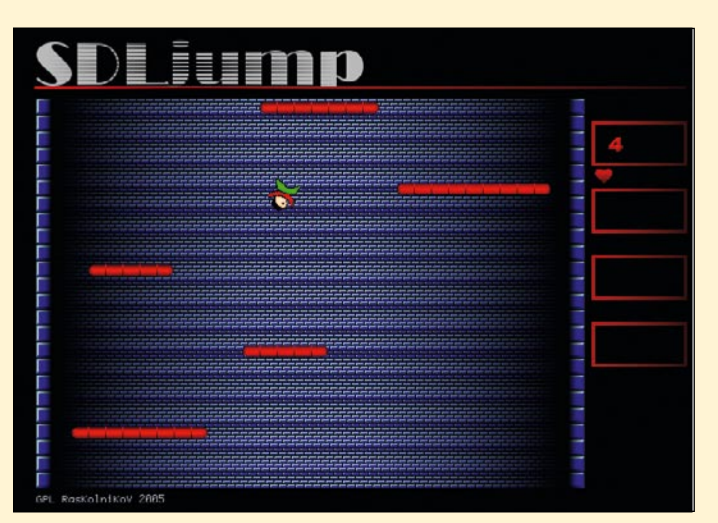


• **ВЕРСИЯ** 0.91 • **WEB** http://sdljump.sourceforge.net

Слово 'jump' (прыжок) в имени игры – определённо залог успеха: вспомните, к примеру, *Jumping Jack* из восьмидесятых или восхитительно-дурашливый *Jumping Flash!* для PlayStation. SDLjump пытается взбодрить классический *Xjump* с помощью новой графики и возможностей, сохранив чудаковатый и опасно-притягательный игровой процесс прототипа. Чтобы собрать игру из исходных текстов, потребуются *SDL* и *SDL_audio* – они присутствуют во всех основных дистрибутивах, а также в каталоге **Essentials** на нашем диске.

SDLjump прост — проще не бывает: надо запрыгнуть на башню. Вашего героя забросили в двумерную среду, прокручиваемую по вертикали, и вы должны скакать наверх по платформам, расставленным на игровом поле. При наличии достаточного времени, больших платформ и управления в стиле *Mario World* это кажется несложным. Но ваш герой отскакивает, катится кубарем и скользит, заставляя вас очень тщательно прицеливаться по мере подъема. За каждую новую платформу, на которую вы приземлились, вы зарабатываете одно очко.

SDLjump добавил кучу желанных обновлений и новые возможности, например, весёлый многопользовательский режим. Два, три или четыре игрока собираются вокруг клавиатуры и после сигнала к прыжкам лезут из кожи вон, чтоб не слететь с экрана. Иногда получается непроходимое месиво из летающих фигурок, но в любом случае попробовать стоит. В *SDLjump* также включена система поддержки тем, так что вы можете перекрасить его в *Xjump* или придать вашему герою вид в стиле Sonic, и плавная прокрутка экрана – ещё один козырь по сравнению с дёргающимся *Xjump*.



Мистера Legsprings'а швыряет по экрану с головокружительной скоростью... Где «преступление», там и «наказание», г-н Raskolnikov (он же автор)!

Несмотря на запредельную скорость перемещений вашего персонажа и на все его выкрутасы вокруг платформ, вы всегда ощущаете контроль над ситуацией, так что игра *SDLjump* превосходна и в режиме одного участника. Падая с платформы, вы отчетливо понимаете, что это результат ваших собственных ошибок и вас никто не надувал. Этот элемент крайне важен, он продлевает жизнь незатейливому игровому процессу, и если игра проигрывает в сложности, она навёрстывает в веселье. Самом бурном.

УТИЛИТА ОБНОВЛЕНИЯ САЙТОВ

Ksitecopy

• **ВЕРСИЯ** 0.1.13 • **WEB** www.hoernerfranzracing.de/kde/ksitecopy.php

Если вы занимались поддержкой web-сайта, вы, наверное, знаете, как утомительно закачивать файлы по FTP, пытаясь синхронизировать ваш сайт с локально сохранённой версией. Например, если вы изменили несколько файлов на машине, на которой занимаетесь разработкой сайта, то вам нужно будет держать где-то список изменённых файлов, чтобы ничего не забыть во время закачки.

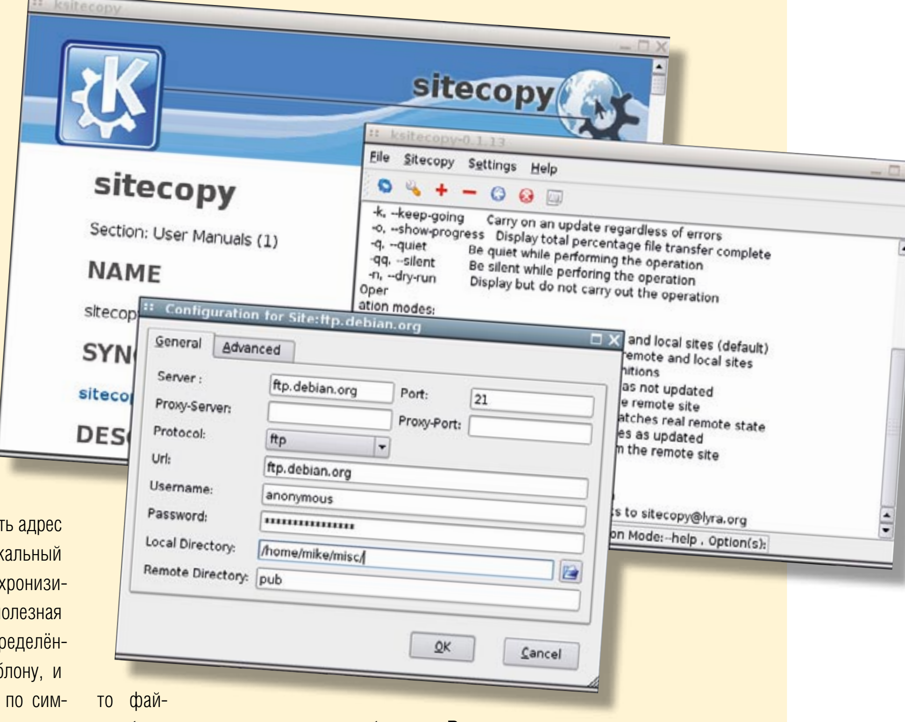
Отнюдь не все FTP-приложения способны облегчить этот процесс, а вот *Ksitecopy* нацелен на сведение данной операции к нескольким щелчкам мыши. Он развился из *Sitecopy* – утилиты командной строки, составляющей основу для операций передачи файлов.

Вам понадобится работающая версия *Sitecopy* – она включена в архив на прилагаемом диске. *Ksitecopy* использует KDE, поэтому придется установить соответствующие пакеты для разработки. *Ksitecopy* находится на раннем этапе развития, поэтому его интерфейс пока не впечатляет, но в нём присутствует мастер, который проведёт вас через базовые настройки, и довольно много диалоговых окон.

Главное окно *Ksitecopy* состоит из набора инструментов для быстрого доступа к операциям из меню, и панели вывода, отображающей результат вызова исполняемого файла *sitecopy*. В диалоге настроек сайта можно назначить адрес и порт сервера, пароли и локальный каталог, с которым будут синхронизироваться файлы сервера. Есть полезная возможность игнорировать определённые файлы, отбирая их по шаблону, и указывать, нужно ли следовать по символьным ссылкам.

Ksitecopy поддерживает только FTP и HTTP (*Sitecopy* предоставляет поддержку еще и WebDAV и rcp), но для большинства пользователей этого хватит. Программа сравнивает файлы на удалённом сервере с вашими локальными копиями и закачивает все изменённые файлы. Если какие-то файлы были локально удалены, они будут удалены и с сервера.

Ksitecopy – простая утилита; ей не помешали бы некоторые косметические изменения и онлайновая документация. Зато она серьёзно экономит время, если вы работаете на несколько сайтов. Да, она только подмножество *Sitecopy*, но и хорошее подспорье для тех, кто некомфортно себя чувствует в командной строке.



Вы можете вручную указать опции *Sitecopy* и запустить man-страницу этой команды.

РЕДАКТОР ДЛЯ ПРОГРАММИСТОВ

Geany

• **ВЕРСИЯ** 0.3 • **WEB** http://geany.uvena.de

Linux хорошо себя чувствует на программистском фронте – у нас есть такие IDE, как *KDevelop*, *Gambas*, *Anjuta* и другие, а также отличный набор инструментов для разработчиков. С другой стороны, многие разработчики привязаны к традиционным текстовым редакторам, например, *Emacs* или *Vi(m)*, отдавая предпочтение их подвижности по сравнению с большими IDE.

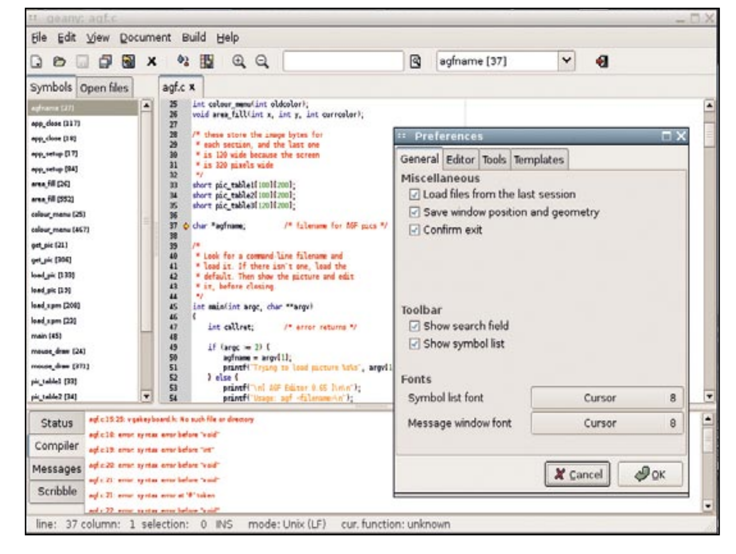
Geany, разработанный Энрико Трёгером (Enrico Troeger), надеется занять среднюю нишу, то есть быть маленьким и быстрым и работать с минимальным набором зависимостей. «Другой целью было максимально отстроиться от специфичной среды рабочего стола вроде KDE или Gnome», говорит Трёгер.

Редактор предоставляет те возможности, которые и ожидаются от среды разработки, оставаясь при этом легковесным. Его главная зависимость – *GTK*, с компонентом редактирования текста *Scintilla*, поставляемом в составе архива с исходными текстами. Основное окно *Geany* изобилует панелями и вкладками: список со вкладками – слева (позволяет выбирать функции в исходном коде или открывать файлы), панель редактирования, тоже со вкладками, – справа, и панель показа результатов компиляции, опять же со вкладками, внизу. Вы моментально переключитесь, куда вам нужно.

Обычные средства редактирования включают: копирование/вставку, поиск/замену и многошаговый откат. *Scintilla* в качестве компонента текстового редактора – отличный выбор: он поддерживает подсветку синтаксиса для всех основных языков, автозавершение кода, нумерацию строк и имеет прекрасную стабильность.

Где *Geany* вступает на почву IDE, так это в пункте меню Build. Он позволяет компилировать исходный код, над которым вы работаете, либо по файлу за раз, либо из нескольких файлов пакетом, через утилиту *make*. Можно запустить получившийся бинарный файл с помощью комбинации клавиш, а также выставить свои собственные параметры (и выполнения, и компиляции). Здорово, что Трёгер включил эти возможности, но не превратил оставшуюся часть редактора в перегруженный возможностями IDE. Команды сборки можно настраивать, как и вид редактора.

Geany – идеальный редактор для начинающих программистов, которые подыскивают нечто поудобнее, чем простой текстовый редактор, но не готовы к мощным IDE. Пишете ли вы программы на C/C++, Perl, *Python* или даже в *Latex*, этот редактор вам пригодится, так как он быстр, надёжен, приятно выглядит и помогает в работе.



Подсветка синтаксиса очень полезна, когда разгрызаешь сложный код...

HotPicks ПОВТОРНЫЙ ВИЗИТ

WEB-ДЕСКТОП

EyeOS

• **ВЕРСИЯ** 0.8.7 • **WEB** www.eyeos.org

Постоянные читатели LXF помнят наш обзор EyeOS в LXF73. Версия 0.7.0.4 впечатлила нас попыткой перенести ОС рабочего стола в web-браузер, что позволит вам подключаться к ней с любого компьютера, имеющего доступ к сети, и оказываться в привычной среде.

Умело сочетая JavaScript и PHP, EyeOS предоставляет простой, но привлекательный десктоп и утилиты прямо в окне браузера, и с момента нашего последнего обзора проведены серьёзные улучшения.

Чисто эстетически, тема по умолчанию в EyeOS выглядит гладко, хотя серое на белом – не самая читабельная цветовая схема. В систему входит небольшой набор утилит, например, текстовый редактор и калькулятор, а также web-браузер, который использует движок своего «родителя», запуская его в окошке-вставке.

Существенные изменения включают поддержку *Internet Explorer* – без этой функции EyeOS на рынке не выжить – и стандартную XML-систему, в которой мини-утилиты могут сохранять свои файлы. Теперь можно управлять несколькими приложениями одновременно, запустив их в каскаде окон. Утилита *EyeHome* действует как точка входа для добавления программ и чтения сообщений, а *EyePhones* представляет собой простую телефонную книгу. Несмотря на то, что местами система ведёт себя немного медленно и слегка странновато, это, несомненно, большая победа.

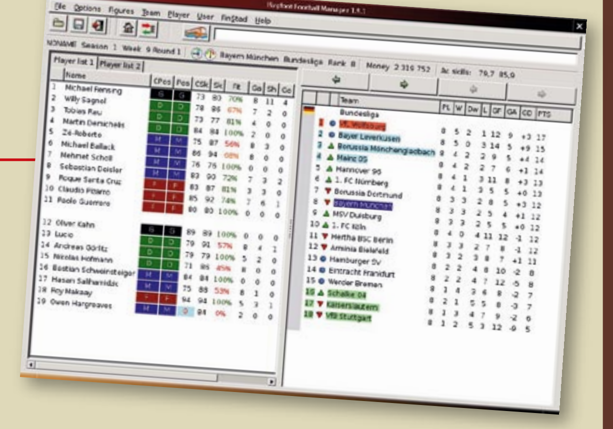
Является ли EyeOS рабочим столом будущего? Очевидно, что с восходом *Gmail* и других web-приложений, мир медленно смещается от жёстких дисков к «сетевым кабелям», используя их в качестве хранилища данных. Чтобы стать адекватной заменой традиционным ОС, EyeOS нуждается в некоторых добавлениях (текстовом процессоре, например), но с появлением релиза 0.9.0, который будет полностью основан на Ajax-технологиях и снабжен онлайновым репозитарием программ, эта система может стать пионером нового подхода к настольным ОС.

Со своим каскадом окон изменяемых размеров, EyeOS создает убедительное ощущение полноценной операционной системы.

ФУТБОЛЬНЫЙ СИМУЛЯТОР

Bygfoot

• **ВЕРСИЯ** 1.9.3 • **WEB** www.bygfoot.com



Недолго осталось ждать момента, когда Чемпионат Мира промчится по планете, а отличным дополнением к нему станет *Bygfoot*. Мы уже делали обзор этой программы управления футбольным клубом в *LXF61*, когда его версия была 1.7.2, и поражались уровнем заботы, которую уделили ей разработчики. Конечно, спортивные игры имеют тенденцию быстро устаревать, поэтому *Bygfoot* успел пережить несколько обновлений, а поскольку он приблизился к версии 2.0, есть смысл взглянуть на него поближе.

К счастью, его единственная зависимость – *GTK 2*, так что для его сборки не придётся рыться в поисках замшелых библиотек (если вы предпочитаете не идти по пути компиляции исходного кода, испробуйте бинарный файл, записанный на прилагаемом диске). Пользователи, знакомые с подобными программами, не встретят никаких проблем, разбираясь в интерфейсе – можно просматривать игроков, предстоящие игры, таблицы лиг и многое другое всего за несколько щелчков мыши. Чтобы вам помочь (или сбить с толку), предоставляется изобилие статистических данных.

Новые возможности и обновления включают: стратегии компьютерных команд, чтобы игроки выглядели немного более естественно; тотализатор, где можно нажить деньгу на матчах вашей команды; смену работы и переход в другую команду; школу для юниоров; спонсирование; и прочее. Вы можете управлять любой командой-участницей на Чемпионата Мира 2006. Разработчики *Bygfoot* весьма расторопны – программа почти полностью переписана с момента последнего обзора и быстро становится зрелой игрой, которую будут уважать фанаты *Premier* и *Championship Manager*. Попробуйте её, чтобы ощутить дух соревнования – и, конечно, управление собственной командой придаст вам уверенности и авторитетности при обсуждениях матчей. **LXF**

«Если у Оуэна Харгривса 0% попаданий – он, наверное, умер?»

НОВЫЕ РЕЛИЗЫ

Свежие и обновленные программы, которые тоже заслуживают внимания

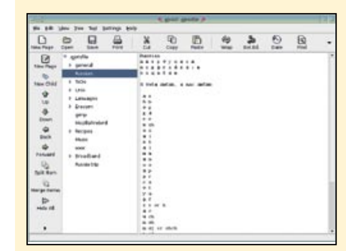
• ***Subversion 1.3.0*** Система контроля версий http://subversion.tigris.org

• ***Rstudio 0.2*** Управление Shoutcast-серверами http://richardsnow.net/BYORS.html

• ***MaraDNS 1.2.00*** Кэширующий DNS-сервер www.maradns.org

• ***Asteroids 1.1*** Классическая игра-клон http://chaoslizard.sf.net/asteroid

• ***Gjots 2.3.0*** Блокнот, использующий Gnome http://bhepple.freeshell.org/gjots



• ***Uade 2.00*** Проигрывает файлы с музыкой Amiga http://zakalwe.virtuaalipalvelin.net/uade

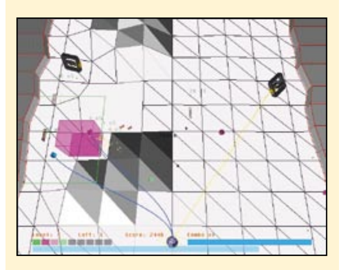
• ***Zim 0.09*** Настольный Wiki http://zoidberg.student.utwente.nl/zim

• ***Jomic 0.9.15*** Просмотр комиксов http://jomic.sourceforge.net

• ***SSXML 0.9*** Легковесный XML-анализатор http://home.earthlink.net/~goatbits/ssxml.html

• ***Cowtacular 0.03*** Инвентаризация и учёт www.happicow.com/cowtacular

• ***Bitfuse 0.11*** Аркадная стрелялка www.cs.umu.se/~ens05aon



• ***xconfig 2.0.0-rc5*** Утилита настройки X http://sourceforge.net/projects/xconfig

• ***Check Website 1.36*** Мониторинг web-сайтов http://checkwebsite.org

• ***RTorrent 0.4.1*** Консольный клиент BitTorrent http://libtorrent.rakshasa.no

• ***Exam4Fun 0.3*** Проверка словарного запаса http://sourceforge.net/projects/exam4fun

• ***AMule 2.1.0*** P2P-клиент для сети EDonkey2000 www.amule.org

Что такое... BOO?



Ник Вейч вглядывается в интригующий новый язык, написанный с прицелом на любителей Python.

>> Это как то связано с привидениями? Тогда должен сказать, что я трусишка…

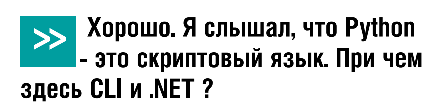
Нет, задача Boo – вовсе не пугать людей. Это безобидная смесь Python и C#, из которой рождается новый язык для среды Mono/.NET.

>> Ой! CLI? C#? Mono? Страшно….

Хорошо, тогда более подробно. Boo - это язык программирования. Я знаю, что вы хотите сказать – их и так предостаточно. Что ж, это верно, но всегда есть причины создать что-то еще. В данном случае, причиной явилось желание Родриго Баррето де Оливейра (Rodrigo Barreto de Oliveira) использовать Python в среде .NET. «Мне все стало ясно», – написал он в манифесте Boo – «Я должен создать новый язык программирования для CLI, который позволит программистам использовать замечательные возможности среды .NET»

>> Хорошо. Я слышал, что Python - это скриптовый язык. При чем здесь CLI и .NET ?

Насчет Python все верно. Это кроссплатформенный интерпретируемый язык, использующийся во многих текстовых и web-приложениях. Он быстро осваивается, а код легко читается, к тому же у него много последователей. Среда .NET была разработана Microsoft как ответ на Java. Идея состояла в создании виртуальной машины с хорошей функциональностью, запускаемой на любой платформе. При наличии соответствующих компиляторов для нее можно программировать на любом языке. Возможно, вы уже знаете аббревиатуру CLI как Command Line Interpreter, но в данном случае она расшифровывается как Common Language Infrastructure. CLI включает в себя описание промежуточного языка (IL – intermediate language) и набора базовых ресурсов, доступных из IL.



> «ВСЕ СТАЛО ЯСНО МНЕ ВПОЛНЕ – ЯЗЫК ДЛЯ CLI СОЗДАТЬ БЫ МНЕ.» АВТОР BOO

>> В чем смысл промежуточного языка?

Намного проще создать компилятор IL для виртуальной платформы и компиляторы высокоуровневых языков в IL, чем полноценные компиляторы для всех языков. Последнее может привести к несовместимости и потребует больше времени.

>> Ага! В таком случае почему бы просто не сделать компилятор Python?

Ну, можно и так. На самом деле уже есть ряд подобных проектов (причем один из них, IronPython, развивается самой Microsoft,-прим.ред.). Но автор Boo хотел использовать преимущества CLI и добавить в Python несколько дополнительных возможностей.

>> Что вы имеете в виду?

Например, статическую типизацию. Python не задает явные типы переменных, он просто присваивает им подходящий тип при записи значения. Это может быть опасно и зачастую приводит к труднообнаружимым ошибкам. Boo использует типы, но он заимствует «утиную типизацию» (duck typing) из Ruby, где 'утке' – типизированной переменной, назначается тип в соответствии с присвоенными данными. Термин происходит от фразы: «Если ходит как утка, и крякает как утка ...». Также можно использовать неявную типизацию, при которой новые переменные создаются как «утки», если не было указано иное.

>> Это похоже на Python с заимствованием из C#, не так ли?

Да, и даже более того. Например, в синтаксисе Python всегда используются отступы для выделения блоков кода. Boo использует это, но он также поддерживает описание структур в скобках, как в C. Пример из документации Boo, вероятно, объяснит все лучше нежели, я:

В стиле Python:

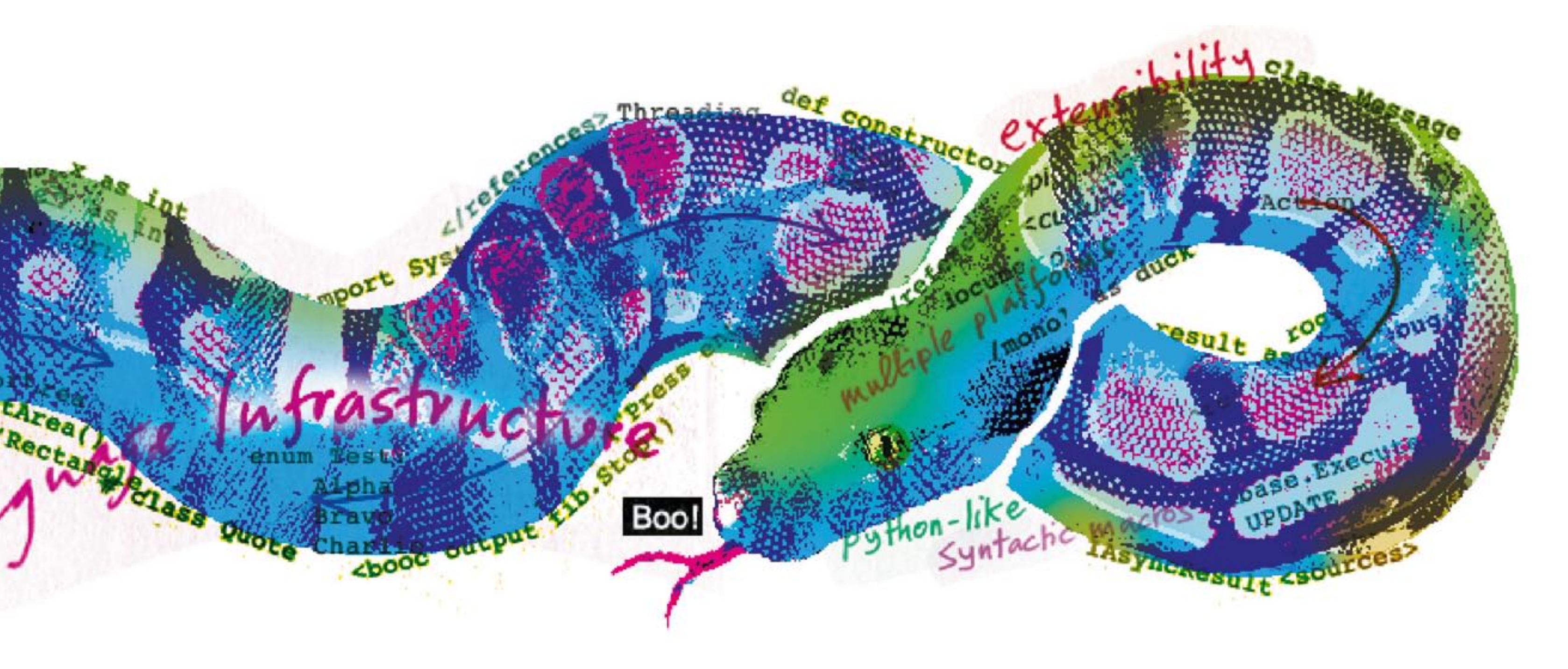
```
button = Button(Text: «Нажми меня»)
button.Click += def ():
		print(«${button} был щелчок!»)
		print(«Да это похоже на другие блоки...»)
```

или с фигурными скобками:

```
button1 = Button(Text: «Нажми меня», Click: {print(«Щелчок!») })
button2 = Button(Text: «Меня тоже!»)
button2.Click = { print(«${button2} была нажата!»);
		print(«внутри {} пробелы игнорируются...»);
		print(«поэтому необходимо использовать точки с запятой для разделения операторов...»);
		print(«но не пишите такой код просто потому, что так можно :)»)
}
```

>> Предположим я хорошо знаю Python, есть ли что-то, чего я должен опасаться?

Нескольких второстепенных вещей – и нескольких важных. Во-первых, нет кортежей, но Boo использует массивы в стиле C#, очень на них похожие. Есть также изменения семантики. Нет нужды использовать 'self' в методах объектов, а стандартные методы классов, такие как **__init__**, **___del__** и **__str___** имеют другие обозначения, типа **constructor()**. Полный список расположен



по адресу: **http://boo.codehaus.org/Gotchas+for+Python+Users**. Но это не так сложно.

Но мне нравятся кортежи...

Что ж, массивы весьма схожи с ними; более того, они сохраняют срезы в стиле Python. С другой стороны, кортежи были идеальны для хранения неизменяемых значений – в системе Boo, значения массивов могут изменяться в зависимости от типа. Ниже приведена запись простой сессии в оболочке Boo (запустите **booish**)

```
>>> array1 = («abc», «ABC», «123»)
('abc', 'ABC', '123')
>>> array2 = array1[1]
'ABC'
>>> print array2[0]
A
>>> print array1[1][2]
C
>>> array1[1][2]=»D»
ERROR: Property 'System.String.Chars' is read only.
>>> array4 = (1,2,3,4,5)
(1, 2, 3, 4, 5)
>>> array5= array4[2]
3
>>> array4[2]=7
7
>>> array5
3
```

Как видно, для некоторых типов значения изменяются. При создании срезов, новые переменные становятся копиями, а не ссылками, поэтому не зависят от изменений в исходном массиве.

Где я могу взять Boo?

Вы можете загрузить исходники и бинарные пакеты с **http://boo.codehaus.org/Download**. Но помните, что для его использования вам понадобятся и другие компоненты Linux.

Например?

Ну, определенно необходима среда Mono (**www.mono-project.com**). Этот аналог платформы .NET с открытым кодом, сейчас поддерживается Novell. Среда Mono содержит виртуальную платформу для запуска вашего CLI-кода. Вам также понадобится *GTK#*, который можно найти там же.

Стоп! Да он тянет зависимости не хуже Gnome! А у меня место на диске не закончится?

GTK# - это не новая *GTK*-библиотека, а набор привязок (bindings), позволяющих разработчикам создавать GUI в стиле *GTK* на *C#*, Boo или других языков. Вы можете найти пакет в своем дистрибутиве или загрузить его с **http://gtk-sharp.sourceforge.net**.

Зачем мне нужна еще и *GTK#*?

Она облегчает создание GUI. Вы можете импортировать эту привязку в ваши Boo-скрипты, также как и в сценариях Python.

```
import System
import Gtk from «gtk-sharp»
Application.Init()
window = Window(«Привет Кнопка»,
DefaultWidth: 100,
DefaultHeight: 150,
DeleteEvent: Application.Quit)
window.Add(Button(«Привет, я кнопка!»,
Clicked: { print(«Кнопка нажата!») }))
window.ShowAll()
Application.Run()
```

Здесь вы можете видеть, что привязки *GTK#* включают классы окон (Window) и приложений (Application). Конструктор для Window говорит сам для себя. Класс Application, в данном случае, нужен только для того, чтобы приложение работало до тех пор, пока не получит событие Quit от Window. Щелчок на кнопке активирует слот Clicked, которые в приведенном примере просто выводит сообщение в *stdout*, но, конечно же, может быть соединен с чем угодно. Итак, вы получили работающую GUI программу в несколько строк кода.

Я не вижу особой разницы по сравнению с Python и *GTK*.

Так и есть - по крайней мере, на уровне кодирования. Истинное отличие в другом. Компиляция данного кода в Mono позволит ему выполняться на всех платформах, на которых работает Mono. Без сомнения, вы можете писать GUI на Python под Windows или Mac, но его создание для Mono делает его более переносимым между поддерживаемыми платформами.

Как писать под Boo? Есть ли IDE?

Вы можете писать программы в любом текстовом редакторе, поддерживающем UTF-8. *Kate*, *Emacs* и прочие прекрасно подойдут. Но вы также можете использовать IDE самого проекта, *MonoDevelop*. Хотя она еще находится в ранней стадии своего развития, это удобный инструмент для работы с Mono, особенно если у вас есть несколько .NET проектов.

Boo это просто последний модный язык-однодневка или он действительно имеет будущее?

Я не могу сейчас предсказать точно, но знаю, что имеется огромное количество людей работающих с Boo и к нему благосклонно относятся многие разработчики, использующие Mono. И конечно же вам следует ознакомиться с множеством примеров, учебников и т. д. на домашней странице: **http://boo.codehaus.org**.

Спасибо. Все не так страшно как я думал.

Без проблем. Да, те постукивания и свист, что вы слышите поздно ночью... Не пугайтесь – это просто я. LXF



Представитель СИЛЫ

Могущественный Google приносит свое состояние и опыт в движение открытого кода. Пол Хадсон беседует с человеком, который помог этому произойти – **Крисом ДиБона** (Chris DiBona)

Визитка LXF

Крис ДиБона

Трудно поверить, но ему всего 33. Он – редактор Slashdot, разработчик игр, администратор VA, консультант по open source и автор в O'Reilly. Его рабочее место – в штаб-квартире Google, Маунтин-Вью, Калифорния

ВОЗРАСТ	33
НАЦИОНАЛЬНОСТЬ	Американец
ИСПОЛЬЗУЕТ UNIX	10
ЯЗЫКОВ ПРОГРАММИРОВАНИЯ	8
КОЛИЧЕСТВО ПК	4
ДНЕВНАЯ НОРМА КОФЕ	0 чашек
ПАР САНДАЛИЙ	1

ОН ГОВОРИТ: «Я люблю open source хотя бы потому, что везде можно наклеить шпаргалки.»

Когда Google прошлым летом объявили одним из спонсоров ежегодного конкурса Open Source Awards от издательства O'Reilly стало ясно, что о невыплаченных призах (бывало...) речь уже не идет. Эта фирма может себе позволить вложить свои деньги куда угодно. В сообществе считают участие Google просто чудесным: у фирмы не только большой карман, она еще и использует открытые инструменты и активно участвует в их разработке. Но есть и скептики: по их мнению, фирма много о себе возомнила и доля ее закрытых программ слишком велика. Крису ДиБона (Chris DiBona), весьма активному менеджеру Google по открытым продуктам, поручили деликатную работу: открыть некоторые приложения Google. Мы встретились с ним, чтобы выяснить, много ли откроется…



Linux Format: **Чтобы мы смогли представить Вас нашим читателям, не могли бы Вы коротко рассказать о своей работе в Google?**

Крис ДиБона: Конечно. Это очень просто. Моя работа – контролировать, чтобы Google работала в соответствии с лицензиями Open Source, дабы не вставал вопрос о корректности использования ее программ. А кроме того, поддерживать контакты с сообществом ПО с открытым кодом, чтобы оно знало – кто мы, что делаем с их программами, как их используем и как можем помочь. Мы делаем немало заплаток и всякого такого для существующих проектов, *Apache*… Ну и я должен заставить все это работать.

Еще мы ведем программы вроде Summer of Code. Мы заинтересовали 419 студентов поработать с открытыми проектами, с *Apache* и Ubuntu по всему миру, чтобы обеспечить постоянный приток разработчиков в наше сообщество.

LXF: **Так Вы почти что миссионер?**

КД: Ну, вряд ли. Google не занимается агитацией – скорее, предоставляет нашим делам говорить за нас. Мы не трезвоним о том, что сделали; пусть нас рекламирует наш продукт.

LXF: ***Apache*** **заполучил немало студентов для Summer of Code, KDE и Gnome тоже…**

КД: Да, а Mono получил 16.

LXF: **Но ведь ядро, наверно, самый крупный открытый проект?**

КД: Не думаю. По-моему, *Apache* крупнее. По части разработчиков… да, это более крупный проект. Я не отказался бы [включить в Summer of Code и ядро]… но с кем мне это обсуждать? Есть ли надежная сторона, которая могла бы оценить этих ребят, и дать добро на вложение в них денег, и сказать, кому платить? Возможно, мы изыщем способы, если снова этим займемся. Но это непросто.

LXF: **Но Novell нанимает немало программистов ядра, пригодных для роли менторов.**

КД: Конечно. Но ко мне они не обращались.

LXF: **Red Hat.**

КД: Они ко мне не обращались. Вот, например, люди из Fedora Core – те обратились. Так что были работавшие над какими-то интересными проблемами с Fedora Core, с Ubuntu, но никто из вышеперечисленных не заикнулся об этом, так что…

LXF: **Действительно жаль, потому что гранты Summer of Code вышли примерно в то время, когда OSDL отставила немало инженеров.**

КД: Ну, мы ведь не брали инженеров, мы работали со студентами. Если ваши читатели хотят разрабатывать ядро как сотрудники, то у нас есть открытые вакансии, у Novell есть открытые вакансии. Возможностей по разработке ядра Linux немало.

О СПОНСОРСТВЕ:

> «Мне нужен именно код, а не просто… деньги. Деньги – это не все.»

LXF: **А сколько досталось ядру?**

КД: Вот тут смешно получилось. Ведь не скажешь: «Эй, разработчики ядра, вам это надо?» У них не как в организации Apache, где для таких дел существует центральный орган. Так что над ядром не работал никто. И потом, ядро не требует приглашений, как, скажем, KDE. Работа над ядром всегда имеет постоянный приток желающих.

С другой стороны, я не выбирал организации, я просто установил сроки, в которые они могли присоединиться, а от разработчиков ядра никто не отозвался.

LXF: **А в итоге оказалось, что с ядром некому работать!**

КД: Знаете, ядро – это еще не все.

LXF: **Ясно, что Summer of Code не было PR-акцией.**

КД: Нет!

LXF: **Тогда что вами двигало – может, это был хитрый ход по набору персонала?**

КД: На самом деле, началось с того, что наши основатели, Ларри (Larry) и Сергей (Sergei), и наш босс Эрик Шмидт (Eric Schmidt) долго переживали по поводу того, что у людей нет средств просто пойти учиться и стать лучшим студентом, лучшим компьютерным специалистом… чем они вынуждены заниматься летом? Отупляющей подработкой, ничего общего не имеющей с компьютерами. Они не развиваются, они тормозят свое развитие в качестве компьютерных специалистов. [Основателям Google] это было ненавистно. И как только я освоился в Google, мне сказали: «Решишь эту проблему? Мы даем тебе $x миллионов в год, и хорошо, если бы ты увязал это с открытым кодом» Так я и сделал, и выкатил эту программу. Вот что двигало основателями.

Кроме того, имеются побочные выгоды – конкурентные, стратегические, всякие. Потому что сильное открытое сообщество означает для всех нас в компьютерной индустрии поле для игры в более высокой лиге, я в это искренне верю. Но это не являлось изначальной целью. Для открытого сообщества это хорошо, а нам это очень нравится. Мы используем тонны ПО с открытым кодом и считаем друг друга товарищами, понятно? Нам нравится работать с открытым сообществом, и это наша форма работы.

LXF: **Где еще, по Вашему мнению, Google может быть спонсором открытого сообщества?**

КД: Вообще-то мне не нравится выражение «спонсорская помощь». Мне не нравится «предоставление субсидий», мне не нравится «окупаемость». Мне нравится выражение «совместная работа». Мы считаем их равными нам в компьютерной сфере, мы не считаем их объектами благотворительности, честно. Забавная штука наблюдается насчет открытого сообщества: за последние пять лет многие из организаций, которые мы с вами считаем выдающимися, получили просто тонны поддержки – финансовой, рабочей, например, через Tigris или SourceForge. Им не нужна спонсорская помощь, им нужно побольше людей – разработчиков кода.

Что делаем мы? Мы открыли кое-какой код на **code. google. com**, постарались раскрутить деятельность по созданию заплаток нашими командами в Google и местными командами за пределами Google, у нас есть люди, которые работают над GCC и над его улучшением, и у нас есть программы типа Лета за Кодом… мы просто выпускаем больше кода.

Потому что опыт показывает: можно помогать, и содействовать, и предоставлять спонсорскую помощь, но есть конкретные преимущества, которые проект получает от самого кода. И они серьезно перевешивают по значимости временную выгоду от таких мелочей, как поездки и прочее.

Так что я не считаю слово «спонсорство» уместным. Поскольку спонсорская помощь подразумевает подчинение. Мы не хотим управлять открытым сообществом, это не к нам.

Должен сказать, меня восхищает, как с этим управляется IBM. По большому счету они никому не навредили: они не пытаются что-то отобрать или поломать, они проделали большую и отличную работу. Мы не собираемся принимать их за образец для подражания в своей деятельности, потому что мы – совершенно разные фирмы, но на самом деле я хочу получить код, мне не нужны просто… деньги. Деньги – это не все.

>>



LXF: Не могли бы Вы уточнить, как именно Google «сотрудничает» с открытым сообществом? Есть Summer of Code, есть Google API...

КД: Мы открываем код. Для создания новых программ у нас в Google есть множество инструментов. Например, у нас есть собственный malloc, есть большое количество средств повышения производительности, есть собственная реализация хэш-таблиц, у нас есть библиотеки на Python, частично работающие по типу Elisp…

И мы открыли все эти инструменты, чтобы каждый мог создавать программы еще лучше. Можно просто открыть для сообщества примеры API и прочее, и это будет здорово, но они не существуют без Google. Мы хотели открыть то, что не просто позволяет лучше работать с нами, но позволяет создавать лучшие программы, и точка.

LXF: А какую лицензию предпочитает Google?

КД: Обычно мы берем BSD, но не чураемся и LGPL. Мы хотим обеспечить людям пользование нашим продуктом. Для этого подходят и GPL и LGPL, но GPL не разрешает использовать эти программы поставщикам проприетарного ПО, поэтому наши инженеры часто выбирают другие лицензии. Выбор лицензии предоставляется инженерам. Если для них это существенно, пусть берут что хотят, нам важнее выпустить код, чем вступать в религиозные войны.

Мы выбираем BSD еще и потому, что нам проще ее отслеживать с точки зрения наших перспектив и нашего уровня работы с персоналом – ее вообще не надо отслеживать. Если вы выбрали GPL, то должны в большей степени следить за программой, а выбрав BSD, вы просто можете заявить: «Вот, берите, наслаждайтесь» – и практически про все забыть.

ПО ПОВОДУ ЗАКРЫТОСТИ PAGERANK

«Дело в том, что есть еще люди, которые ежедневно пытаются разгадать индекс и заняться поисковым спамом.»

LXF: Какая часть программ, разработанных внутри Google, попадает в открытое сообщество?

КД: Ну, пока не много. Мы же только начали открывать код. Но каждый месяц мы открываем чуть больше. Немало наших инженеров что ни день приходят ко мне и говорят: «Слушай, я бы хотел выпустить вот это». Объясняю, как это сделать, проверяю наличие файлов Readme в авторском стиле, наличие лицензии, чтобы все было, как положено. Удостоверяемся, что у юристов не будет к нам претензий, и – открываем.

LXF: И через четыре года Google станет на 100% открытой фирмой?

КД: О, нет. Никоим образом. Некоторые вещи мы не можем открыть, потому что они либо лицензированы, либо открывать их бесполезно.

Например, некоторые наши программы просто невозможно использовать, если у вас нет вычислительного центра этак на сотню компьютеров. Или если ваш компьютерный центр не в точности повторяет архитектуру нашего. Кончится тем, что вы просто угробите время на переделку программы, а это совершенно бессмысленно. Если нет реальной возможности согласованного широкого использования программы, то ее не стоит открывать. Потому что на открытие нужно время. Желающим открыть какой-то продукт я бы посоветовал открывать то, чем люди смогут воспользоваться немедленно, а не идти сначала покупать компьютерный центр, как у нас, и оборудование, в точности как наше.

И потом, есть еще кое-что. Мы не собираемся открывать PageRank [собственная система ранжирования web-страниц в поисковом индексе Google] и все подобные программы, потому что открыть их означает их разрушить. Если рассказать, как происходит сортировка, кто попало будет дурить поисковую машину, нарушая порядок, и поиск через Google станет бессмысленным. Нам этого не надо.

LXF: Итак, это обречено на секретность?

КД: Можно и так сказать, но это неправда. Нет. Дело вот в чем. Безопасность презирать легко, но вы, значит, не разбираетесь в рынке Web – надеюсь, это не очень грубо звучит. Суть в том, что есть люди, ежедневно пытающиеся разгадать индекс и заняться поисковым спамом, и если вы облегчили им работу, значит, вы проиграли войну. Это постоянная битва, нам приходится ее вести, чтобы обеспечивать качественный результат поиска. Мы считаем, что хорошо с этим справляемся, а если мы откроем данные функции, у нас возникнут сложности.

Мне также не хотелось бы открывать код, который не находится в активной разработке инженерами Google. Наши инженеры очень заняты, и если кто-то захочет открыть код, на который у них нет времени, получится, что мы просто кинули его за забор, от этого мало пользы. Хотелось бы все-таки открывать код таким образом, чтобы сторонние разработчики не вешались от тоски.

LXF: То есть иметь возможность обеспечения долгосрочной поддержки вашего продукта?

КД: Именно.

LXF: Вы упомянули, что Google широко использует открытый код в поисковой машине. Какой код?

КД: Мы задействуем многое из того, что вы называете Linux. Двоичные инструменты ядра Linux, набор для компиляции, Perl, SSH, SSL, всякое такое. *Apache* мы не особо используем, хотя это многих удивит – у нас есть собственные web-серверы. Зато используем многое другое.

Много пользы мы извлекаем из ядра. Это фантастика – располагать ядром с открытым кодом, потому что если мы хотим внести изменения, мы их просто вносим, понимаете? Если хотим узнать причину какой-либо странности, то просто выясняем эту причину, поскольку можем посмотреть

код. Это фантастика. Еще мы пользуемся *TomCat*, ну и разными другими проектами.

LXF: **Ваши инженеры передают свои наработки проектам, по мере надобности?**

КД: По мере надобности, и отчасти как сами захотят. Мы стараемся, чтобы им этого хотелось почаще, потому что нас всех радует выход новых заплаток. Так делается. Просто на это нужно время.

LXF: **А Вы не опасаетесь, что придется поддерживать длинный список заплаток для ядра?**

КД: Для ядра – да, потому что имеется множество наших заплаток, которые основному ядру не нужны. Я не шучу. Больше того, и поддерживать их незачем. Среди них есть весьма капитальные изменения в ядре Linux специально для нашей среды, многое – для сетевых программ, и тому подобное. На самом деле, мы… когда нам нужно развивать эти вещи, Уолт Драммонд (Walt Drummond), глава этой группы и я работаем вместе, чтобы проверить, где могут быть полезны новые заплатки, включая не очень чистые и вообще ужасные, иногда даже просто выносим их на сайт Google, хотя команда разработчиков ядра их никогда не примет. Но мы все равно их открываем. Людям, которые захотят заниматься тем же, чем и мы, они могут пригодиться.

LXF: **А может, делать все как следует – производить изменения в хорошей, ясной…**

КД: Вы подразумеваете, что «как следует» для нас и для вас – одно и то же. Но в этом и прелесть открытого кода, верно? Мы сами можем определять судьбу своего программного продукта. То, что подходит нам, вполне может оказаться для вас неправильным. У вас ведь не обязательно такой же профиль, как у нас, и от ядра вам потребуется другое.

LXF: **Мне кажется, что открытие заплаток, как минимум, позволяет, например, Red Hat сказать: «Э, да у нас есть версия AS, которая подойдет некоторым потребителям» – или дает им возможность ее отвергнуть.**

КД: Да, конечно, мне хочется создать побольше возможностей. А еще – хочется рассказать, чем мы занимаемся, чтобы те, кто интересуются ядром и собираются перейти на Linux, могли сказать: «О, даже Google этим пользуется», понимаете? В этом есть некая сила, и это отлично – пусть люди знают, что Google с удовольствием использует открытое ПО. Потому что так и есть. И, может быть, кто-то изменит свое мнение о программах с открытым кодом и тоже начнет их использовать.

LXF: **Теперь не про открытый код: есть несколько серьезных попыток взлома Google Maps API. Например, берешь карту Чикаго и видишь районы проституции или типа того. Можно найти криминал в любом городе, где угодно – просто блеск!**

КД: Да, крутая штука.



LXF: **Как Вам кажется, по мере того, как Google тянет когти к…**

КД: Google, по-вашему, чудовище?

LXF: **Ох, простите, по мере того, как Google расширяется в новые сферы, больше станет программ вроде PageRank, чей код не будет открыт? Как насчет Google Maps?**

КД: Штука вот в чем: многие не понимают, что Maps нам самим стоит денег. Мы не просто берем данные у правительства, а используем коммерческие базы данных, за них нужно платить. Поэтому пришлось выбрать компромисс – позволить использовать наши карты на других сайтах, это очень необычная практика. Обычно, когда делают API карт, людей отсылают на свои сайты. Но мы хотели попробовать дать на них людям право собственности, чтобы они использовали их именно так, как им надо. Так что Maps – это сплошные контракты и сделки, юридические тонкости, тут действительно есть ограничения на пользование. Эту программу нельзя открыть, как другие.

И вот еще чего многие не понимают: создавая что-то в Google, вы имеете огромную прослойку инфраструктурного кода, обеспечивающего web-сервисы и прочее. Поэтому чтобы открыть все те вещи, которые, по-вашему, и есть Google, нам придется открыть и все остальное.

LXF: **Вы хотите сказать, что все взаимосвязано.**

КД: Именно так. Но поэтому-то и здорово работать в Google. Внутри фирмы вы можете создать сервис и моментально его запустить, если вам это надо, вот почему лаборатории Google такие классные.

Если кто-нибудь скажет: «Хочу запустить версию Moon Maps [Карты Луны, – прим. перев.] » – пожалуйста, такое ПО у нас уже есть. Запустить новый сервис нетрудно, если у вас есть действительно хорошая идея. LXF

Крис ДиБона говорит о Лете за Кодом (Summer of Code), серверах, и не говорит о проституции в Чикаго на **www.linuxformat.co.uk/mag/dibona/html**. Не пропустите!

Знакомьтесь: FreeBSD



Linux – это не единственное «лакомство» для тех, кому нравятся Unix-подобные операционные системы. FreeBSD бесплатна, стабильна и открыта. **Майк Сондерс** (Mike Saunders) следит за её развитием и предлагает руководство по установке для тех, у кого возникнет желание попробовать ее в работе...

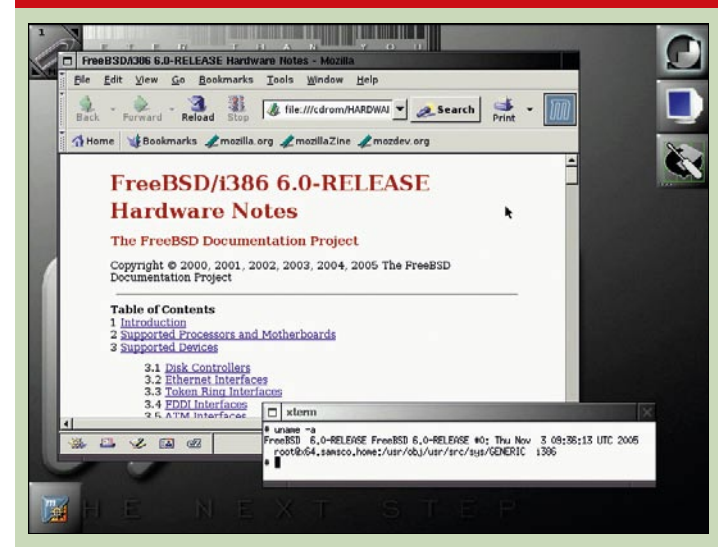
FreeBSD была одной из заветнейших тайн компьютерного сообщества. Эта открытая Unix-подобная операционная система существует с 1993 года, но лишь недавно проект пересмотрел свой сайт, запустил релиз новой линейки и (наконец-то) удостоился внимания со стороны СМИ. Больше десяти лет компьютеры с FreeBSD тихо трудились как супер-надежные серверы и рабочие станции, но их преимущества затмевались взрывным распространением Linux. Теперь, с выходом последнего релиза 6.0, FreeBSD готова предстать во всей своей красе.

Путь к этой версии был тернистым. Во FreeBSD 5.0 были внесены радикальные изменения и обновления системы (например, улучшенная поддержка многопроцессорности и потоков), но программистов подвели амбиции: FreeBSD временно потеряла свою знаменитую стабильность, и многие пользователи предпочли остаться на несокрушимой 4.x. Ветвь 5.x приобрела клеймо неустойчивой – анафема для пользователей BSD. Новый старт с линейкой 6.x стал для разработчиков шансом восстановить былую репутацию, которую FreeBSD завоевала пусть у небольшой, но преданной армии поклонников.

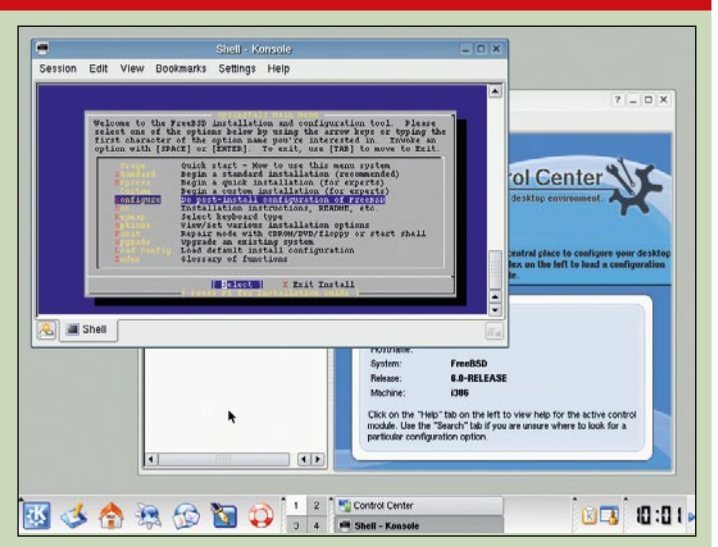
Впрочем, не только ошибки отразились на развитии FreeBSD. Из-за разногласий с коллегами один из наиболее талантливых FreeBSD-программистов (приверженцы Amiga знают его как человека, стоящего за компилятором Dice C), Мэтт Диллон (Matt Dillon), покинул проект и занялся Dragonfly BSD – ответвлением 4.x. Это подорвало настрой FreeBSD-сообщества, и встал вопрос о разумности традиции вести работу над открытым проектом закрытой командой разработчиков. Естественно, и ошибки и потеря ведущего разработчика не прошли бесследно для истории, опыта и уважения, которые FreeBSD приобретала долгие годы. Проект был отброшен назад.

Про Linux мы знаем, что он возник на базе проекта GNU; а где зародилась FreeBSD? На самом деле, в Калифорнии. Разработанная в Университете города Беркли версия системы Unix – Berkeley Software Distribution (BSD) – была весь-

КАК Я ВЫГЛЯЖУ? РАБОЧИЙ СТОЛ FREEBSD



За рабочим столом FreeBSD вы будете как дома: большинство приложений Linux здесь работает. На рисунке – *Window Maker*.



KDE тоже добавляет шарма к FreeBSD – загляните на http://freebsd.kde.org и узнаете последние новости и советы по установке.

ма популярна в университетской среде в 70-х годах. Но в ней содержалось немало строк исходного кода Unix, права на который принадлежали AT&T, так что в конце 80-х разработчики приступили к «чистке», заменяя проприетарные фрагменты открытым кодом.

К несчастью, в 1992 году AT&T затеяла судебный процесс против BSDi – компании, которая портировала BSD на обычные ПК архитектуры x86. Соглашения они достигли без суда в 1993 году – абсолютную законность коду BSD придали удаление всего трех файлов и пригоршня изменений. Однако ущерб был нанесен: хотя Интернет добрался до широких масс и концепция свободного ПО набирала обороты, немногие разработчики решались связываться с кодом, сомнительным (как им казалось) с точки зрения закона. Как следствие, центр внимания переместился к Linux. Потребовалось много лет, чтобы разработчики преодолели дискомфорт, вызываемый BSD.

FreeBSD возникла как наиболее успешный порт BSD для ПК, вместе со своими «кузинами» – NetBSD и OpenBSD, нацеленными на переносимость и безопасность соответственно. FreeBSD как ОС ориентировалась и на рабочие станции общего назначения, и на серверы; высокая степень совместимости с другими разновидностями Unix позволяла ей запускать большинство программ, предназначенных для Linux, однако акцент делался на сплоченное сообщество разработчиков и длительное тестирование новых функций. Похвальная цель, особенно учитывая, что разработка Linux в середине девяностых велась в стиле Дикого Запада – «хватай-кто-успел».

Два миллиона причин

Развитие Linux идёт семимильными шагами, но почему FreeBSD всё ещё привлекает разработчиков и пользователей (обеспечивая, по последним оценкам, работу двух миллионов web-серверов)? Факторов, способствующих данному выбору, немало: философия разработки, инженерный опыт, общие стандарты... зачастую дополнительным аргументом для многих программистов и пользователей является её лицензия (см. врезку *«Лицензия BSD» на стр. 63*).

И, конечно, нельзя не упомянуть легендарную стабильность FreeBSD: её использует Yahoo! (один из крупнейших интернет-порталов), и долгие годы именно она обеспечивала работу Hotmail. Кроме того, из нее пришел ряд пользовательских инструментов в Mac OS X компании Apple. FreeBSD хорошо известна своим консервативным по»дходом к свежеизобретенным функциям. Вместо того, чтобы «выкатить» новый код, а уж потом работать над его стабилизацией, команда FreeBSD ищет долгосрочные решения – даже если решение приходит не вдруг. Релизы с проблемами, вроде ранних версий ветви 5.x, крайне редки. Конечный результат – «вылизанная» и непротиворечивая система, качество которой заметно во всем, от аккуратных справочных страниц до надежного как скала ядра. В итоге, администрирование системы FreeBSD очень эффективно. «Мастера конфигурации» здесь не водят вас за ручку, но это с лихвой компенсируется логичной и предельно понятной документацией.

Плюсы и минусы

Одна из привлекательных особенностей FreeBSD – её единство: одна система, одно дерево исходного кода, один инсталлятор, одно руководство пользователя, один формат пакетов, один набор сценариев инициализации. Когда вы изучаете FreeBSD, вы изучаете операционную систему, а не просто одну из ее разновидностей: вам не придётся иметь дело с аллергией на отдельно взятые дистрибутивы. Однако Linux всё ещё лидирует в ряде областей (прежде всего на рабочих столах пользователей), обеспечивая более качественную поддержку современной комплектации оборудования домашних ПК, а также коммерческих приложений. (Во FreeBSD поддерживается двоичная совместимость с Linux, но она не идеальна). Кроме того, по Linux имеется гораздо больше руководств для начинающих пользователей.

Несмотря на изначальную обособленность, FreeBSD пользуется некоторыми разработками GNU, например, компилятором *GCC*. Что касается ПО, по нашим оценкам, 95% открытых программ, обычно используемых в Linux, замечательно работают и на FreeBSD – *Apache*, *MySQL*, KDE, *Gnome*, *Firefox*, *OpenOffice.org* и тысячи других. Удобная система портов во FreeBSD, которые обеспечивают скачивание программ, установку необходимых «заплаток» и их компиляцию при минимуме нажатий на клавиши, настолько удачна, что оказала сильное влияние на систему портов *Portage* в Gentoo. При повседневной работе в качестве сервера или рабочей станции FreeBSD и Linux очень похожи: запустите Bash и KDE, и вы практически не почувствуете разницы. Администрирование системы и вопросы разработки, однако, отличаются.

В этом месяце мы поместили на наш диск FreeBSD 6.0, и на следующей паре страниц покажем, как установить и использовать эту впечатляющую ОС. Сперва ознакомьтесь со врезкой «*Об инсталляторе FreeBSD*», а потом приступайте собственно к инсталляции.



ОБ ИНСТАЛЛЯТОРЕ FREEBSD

FreeBSD запросто соорудит сервер из старого «железа» – достаточно процессора 486 и 24 МБ оперативной памяти. Для «настольного» использования потребуется по крайней мере 300 МГц-процессор и 128 МБ ОЗУ. Перед началом установки скопируйте или распечатайте файлы **HARDWARE.TXT** и **INSTALL.TXT** с нашего диска. Вероятно, они вам не потребуются, но если что-то пойдёт не так, с ними можно проконсультироваться насчёт дальнейших шагов.

Перед тем, как ставить FreeBSD, поговорим о системном инсталляторе – *Sysinstall*. *Sysinstall* почти не менялся уже несколько лет – благодаря диалогам и меню он удобнее инсталлятора OpenBSD, основанного на интерфейсе командной строки, но ему далеко до дружественных графических инструментов, обычно используемых в Linux. Тем не менее, он делает работу чётко и надёжно, не злоупотребляя ресурсами, поэтому инсталляция выполняется очень быстро. *Sysinstall* выполняет разбиение жёсткого диска и установку ПО вместе с некоторыми настройками (например, раскладкой клавиатуры и запуском сервисов). Последующая конфигурация выполняется из командной строки при «моральной поддержке» замечательно подробного руководства. Думая, что это очень похоже на установку Debian или Slackware, вы будете правы.

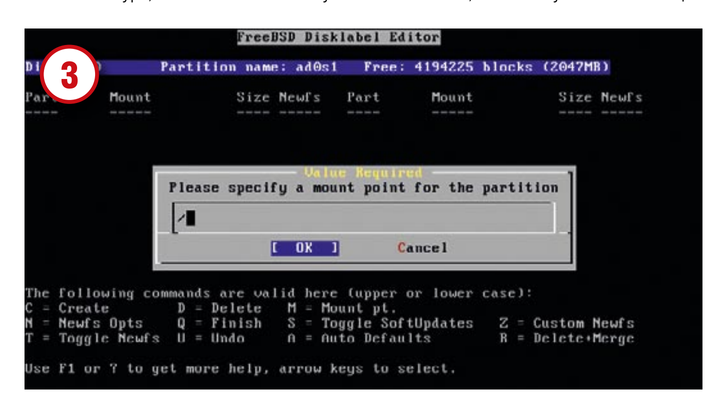
<<

КАК УСТАНОВИТЬ FREEBSD

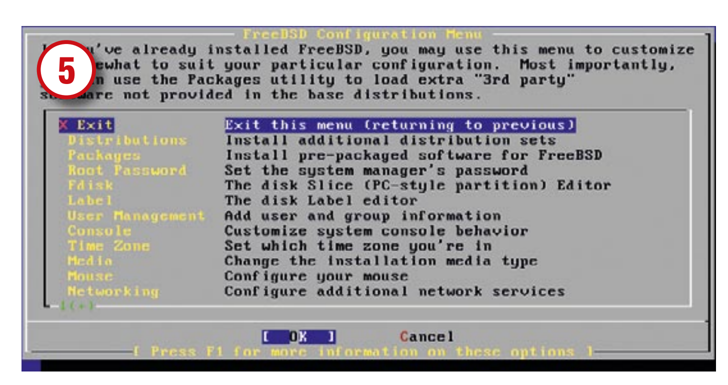
Получите самый отказоустойчивый компьютер, который и за деньги не купишь!



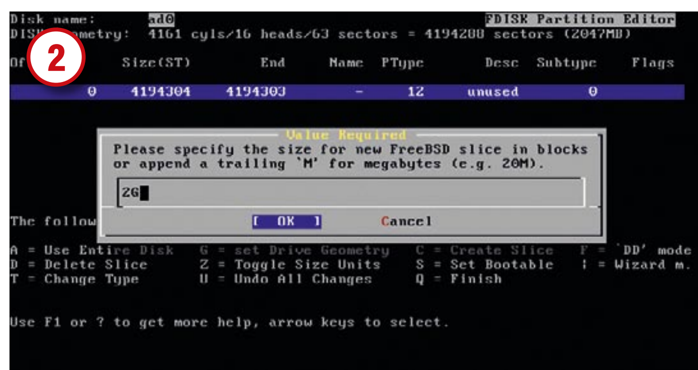
Итак, начнем. (Заметьте, что FreeBSD требует технических знаний, так что новичкам в мире Unix советуем начать с Mandriva или SUSE). Вставьте диск в ваш привод, перезагрузитесь и проверьте, что ваш компьютер настроен на загрузку с CD/DVD. (Возможно, вам придется войти в BIOS, нажав F1, F2 или Del в начальный момент загрузки, и изменить порядок поиска загрузочных устройств). Возникнет экран, показанный выше – если у вас USB-клавиатура, нажмите 7. В ином случае нажмите Enter, чтобы запустить инсталляцию.



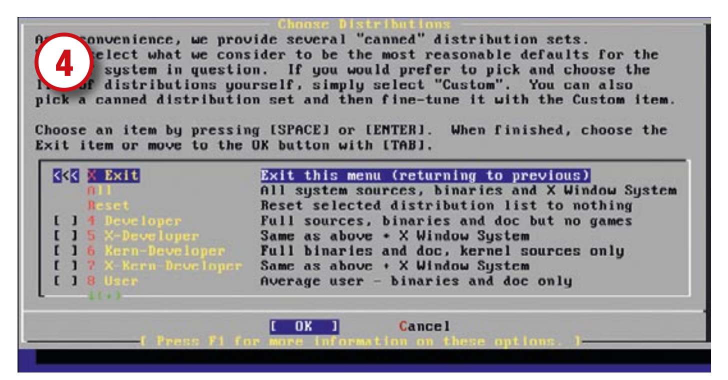
Выбрав установку загрузчика *BootMgr* в очередном меню (*Lilo* или *Grub* можно будет вернуть позже), вы увидите этот экран. Мы снова создаем разделы, на этот раз – разделы FreeBSD внутри только что созданного основного. Нажмите «C» для создания файловой системы выбранного размера, монтируемой в корень (/), затем выделите её и нажмите «S», чтобы включить SoftUpdates (в BSD это аналог журналирования). Если хотите, можете также создать раздел подкачки (swap). Закончив, нажмите «*Q*». [Во FreeBSD принято выделять как минимум 3 файловых системы – /, /usr и /var, что повышает надёжность и гибкость. Можно выполнить разбивку по умолчанию, просто нажав «A», – прим. перев.]



Разделавшись с копированием файлов, инсталлятор задаст вам несколько вопросов об основных настройках – просто последовательно отвечайте на них (хорошая идея – включить режим двоичной совместимости с Linux, можно будет запускать проприетарные Linux-приложения вроде *RealPlayer*). После чего вы сможете вернуться на этот экран и изменить какие угодно настройки; теперь нажмите Exit и Exit Install для перезагрузки (не забудьте вынуть из привода инсталляционный диск). Поздравляем: вы на пороге мира FreeBSD!

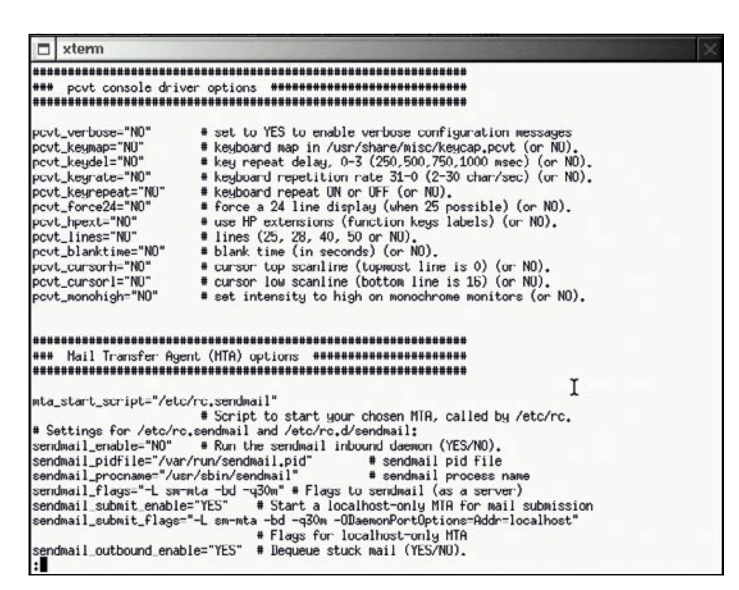


Вслед за сообщениями ядра появится инсталлятор *Sysinstall*. Воспользуйтесь клавишами управления курсором и Enter и выберите стандартную установку (пункт «Standard install». Вы попадёте на экран разбиения диска, где нужно будет создать «участок» (slice) для FreeBSD. (Участок – это просто раздел жёсткого диска [partition в терминологии Linux, – прим. перев.]). Нажмите «C», чтобы создать раздел, задайте размер (в идеале – не менее 1 ГБ) и оставьте тип по умолчанию. Выберите его и нажмите «S», чтобы сделать его загружаемым, затем – «Q» для выхода.



Перед вами экран выбора ПО. FreeBSD различает базовую систему и вспомогательные пакеты, так что вы не найдёте здесь *Firefox* со товарищи (всё это будет, но позже – на этапе выбора пакетов). Отметьте X-User и All, и когда инсталлятор спросит, не поставить ли систему портов (Ports), ответьте «Yes». Также вы можете получить вопрос о выборе дополнительного программного обеспечения. Покончив с этим, выберите в качестве носителя, откуда будет выполняться установка, CD/DVD, и *Sysinstall* приступит к копированию файлов на ваш жёсткий диск.

После перезагрузки, войдите в систему как root (или как любой из пользователей, которых вы создали в процессе инсталляции) и осмотритесь. По умолчанию предусмотрена командная оболочка с минимумом возможностей, но если вы ставили *Bash*, то он находится в каталоге **/usr/local/bin**. Вы сможете принять его на постоянную работу командой *chsh*. Для возврата к экрану настроек введите **sysinstall** и выберите пункт Configure. Наслаждайтесь! И читайте дальше для получения информации о пользовании системой.



К счастью, большинство настроек FreeBSD сосредоточены в одном файле – хорошо прокомментированном /etc/defaults/rc.conf.

Советы по настройке

Если вы планируете использовать FreeBSD на рабочем столе, то наверняка захотите запустить *X Window System*. Это можно сделать несколькими способами: команда **X -configure** автоматически создаст конфигурационный файл, который, возможно, придётся слегка подправить; **xorgcfg** настраивает «иксы» с помощью графического интерфейса; наконец, когда все остальное не работает, спасательным кругом послужит текстовая утилита **xorgconfig**. Если вы не уверены, что знаете характеристики вашего оборудования, выбирайте графический драйвер VESA: он не слишком быстр, зато работает практически на любой машине. (Учтите, что мышь PS/2 во FreeBSD – это устройство **/dev/psm0**).

Наверное, вам захочется установить и другие полезные программы. Вставив наш диск, наберите **sysinstall** (от имени root) и выберите Configure > Packages. Теперь установите дополнительные программы с CD/DVD – перейдите в меню All, отметьте *Bash* (эта оболочка удобна для работы) и выберите оконные менеджеры и приложения, какие нравятся (их распаковка обычно идет в директорию **/usr/local**). Если у вас есть широкополосное подключение к Интернету, вы сможете выбрать гораздо больше программ, указав в качестве источника HTTP или FTP.

Следующий этап – небольшая настройка базовых функций.

Взгляните на файл **/etc/defaults/rc.conf** – это основной конфигурационный файл, определяющий, что происходит с системой во время её загрузки. Все настройки документированы; если нужно что-то изменить, лучше всего переписать **/etc/rc.conf** (а не редактировать непосредственно **/etc/defaults/rc.conf)**. Вы можете использовать клавиши Alt+Fn для переключения виртуальных терминалов – очень полезная возможность читать конфигурационный файл в одном терминале, а редактировать в другом.

«СИСТЕМА ПОРТОВ ДАЖЕ ЗАГРУЗИТ И СОБЕРЕТ ЗАВИСИМЫЕ ПАКЕТЫ.»

Если на этапе выбора пакетов вы не позаботились о текстовом редакторе, единственно возможным будет Vi.

Зайти дальше

Куда вы двинетесь теперь, зависит только от вас – если вы планируете использовать FreeBSD как сервер, можете заняться безопасностью вашей машины, установить серверное ПО и подстроить производительность. Если вам нужна настольная система, займитесь поддержкой 3D-графики, пересоберите ядро с поддержкой дополнительного оборудования и настройте уровень совместимости с Linux. Всё это, да и многое другое, отлично разъяснено в исчерпывающем руководстве пользователя (FreeBSD Handbook), копию которого вы найдёте на нашем диске (см. **handbook.html**).

Система Ports заслуживает особого внимания. Если вы установили её при инсталляции ОС, вы найдёте огромное дерево каталогов в **/usr/ports**, содержащее почти все существующие открытые приложения. Здесь нет файлов программ – только скрипты Makefile и примочки, обеспечивающие автоматическую сборку и установку. Просто перейдите в каталог требуемой программы (например, **cd /usr/ports/x11-wm/xfce4**) и введите команду **make install**. Ports сама скачает исходный код программы, наложит специфические для FreeBSD заплатки, скомпилирует код и выполнит установку. Это фантастически просто – даже необходимые зависимости скачаются и установятся без вашего участия. Мы поместили на наш диск несколько собранных пакетов из коллекции портов, кроме того, вы можете «вытянуть» их из Сети, например, так: **pkg_add -r xfce4**.

FreeBSD имеет активное Интернет-сообщество. Обновлённый сайт проекта – **www.freebsd.org** – предоставляет доступ ко множеству списков рассылки и руководствам [в т.ч. и переведённым на русский язык, – прим. перев.] и является превосходным местом для получения помощи. Взгляните на последние выпуски программ для FreeBSD на **www.freshports.org**, а чтобы поддержать проект, можете купить «коробочную» версию на **www.freebsdmall.com**. Различные дискуссии проходят на сайте **www.bsdforums.org** [см. также всеобъемлющий русскоязычный ресурс: www.bsdportal.ru, – прим. перев.], ну и мы тоже не возражаем против вопросов на засыпку о FreeBSD на форуме журнала... LXF

ЛИЦЕНЗИЯ BSD



Соберите в одной комнате нескольких пользователей BSD и Linux, скажите слово «лицензия» – и сбегите. Когда вы вернётесь, увидеть что-то, кроме дымящейся воронки, будет большой удачей. Причиной такого противостояния этих двух сообществ является различное понимание термина «свободный»: с точки зрения GNU и Ричарда Столмана (Richard Stallman), он означает свободу доступа к коду и распространение этой свободы. А вот разработчики BSD считают, что каждый может делать с исходным кодом всё, что заблагорассудится. Короче, в лицензии BSD говорится: «Делайте с кодом всё, что хотите, только не говорите, что это вы его написали».

В противоположность этому, GPL явно запрещает кому-либо помещать код в проприетарные продукты без публикации сделанных изменений. Легко видеть, что лицензия BSD наиболее привлекательна для традиционного рынка ПО, хотя многие сторонники Open Source не испытывают комфорта при мысли, что код, который они разработали, может оказаться закрытым в проприетарном продукте. Их первоначальный код, распространяемый по BSD-лицензии, по-прежнему останется свободным – но это не гарантирует свободу доступа к любой версии этого кода.

Примеры вы можете встретить в Microsoft Windows. Запустите программу *ftp* из командной строки, и вы увидите упоминание Университета Беркли (где возникла BSD). Microsoft может использовать код BSD в своих сетевых утилитах, но никакой благодарностью разработчикам не связана.

Мы не ставим перед собой задачу решить, какая же из двух лицензий лучше – BSD или GPL: просто радуйтесь, что выбор есть.

НОВОЕ В ВЕРСИИ 6.0

Если раньше вы уже работали с FreeBSD и хотите узнать, почему столько шума поднято вокруг 6.0, здесь вы найдете краткий отчет о новых функциях, добавленных в этот релиз:

- Улучшена производительность файловой системы и доступа к диску. В файловой системе появилась улучшенная поддержка многопоточности и многопроцессорности.
- Теперь поддерживаются процессоры PowerPC, хотя и в статусе «экспериментальной» платформы.
- Вы найдёте улучшенную поддержку специальных функций, присутствующих в ноутбуках IBM, Fujitsu и Sony.
- Появилась более широкая поддержка беспроводных сетей, а также протокола безопасности WPA.
- Улучшена безопасность благодаря новой технологии распределения памяти ядра – memguard.
- Оптимизирован планировщик ULE, что улучшило время отклика системы.
- Добавлена возможность шифрования диска – GEOM_ELI.



BSD НА ТВОЕМ

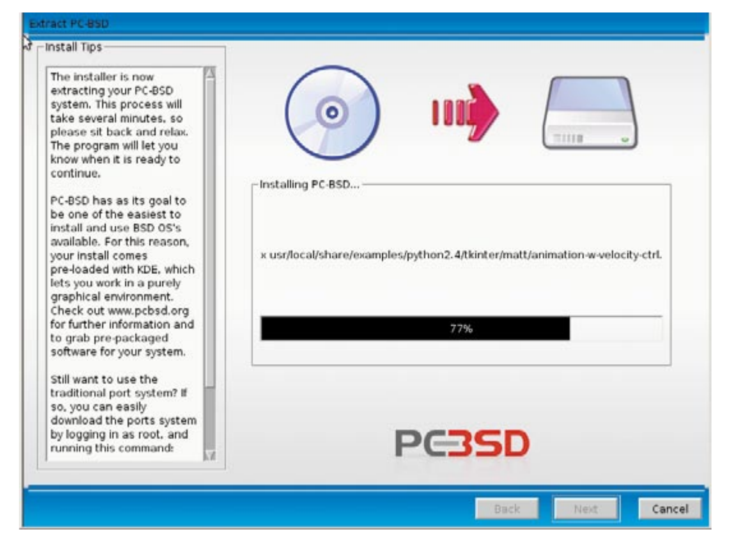
Операционная система *FreeBSD* отличается высокой стабильностью и занимает достойное место на серверах. Однако на рабочем столе пользователя встретить эту красавицу почти невозможно, и виной всему – пугающий новичков текстовый режим инсталлятора и необходимость вручную настраивать графическую подсистему, – считает **Сергей Супрунов**.

Фанаты *FreeBSD* постарались исправить эту несправедливость, и на свет появились несколько проектов дружественных пользователю дистрибутивов. Наиболее яркими и успешными, пожалуй, являются системы *PC-BSD* и *DesktopBSD*. Оба проекта сравнительно молоды (на момент подготовки данной статьи и тот, и другой обещают скорый выход версии 1.0, находясь на этапе предрелизного тестирования), но, тем не менее, им уже есть чем порадовать своих пользователей.

Общие особенности

Оба дистрибутива имеют ряд схожих черт. И в том, и в другом случае на базовую систему не налагается никаких ограничений – вы получите в своё распоряжение «чистокровную» *FreeBSD*, со всеми её возможностями, достоинствами и недостатками. Разработчики дистрибутивов всего лишь сделали за вас часть работы по предварительной настройке системы и упростили процесс установки.

Как *PC-BSD*, так и *DesktopBSD* предлагают пользователю рабочий стол на базе KDE. Если вы сторонник Gnome, то, видимо, вам придется всё-таки иметь дело с «чистой» *FreeBSD*. Теперь давайте посмотрим, чем же эти дистрибутивы отличаются…



Инсталлятор *PC-BSD*: простенько и со вкусом.

PC-BSD

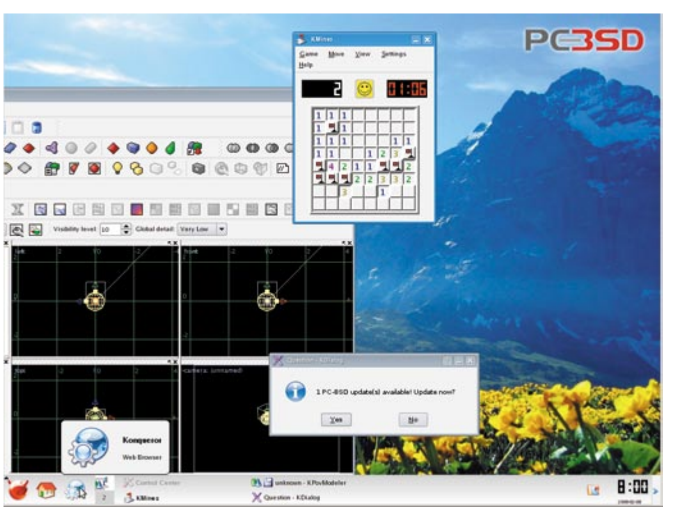
- **САЙТ ПРОЕКТА:** www.pcbsd.org
- **ТЕКУЩАЯ ВЕРСИЯ:** 1.0RC2 (1CD + 1CD с файлами локализации)
- **ОСНОВАН НА:** FreeBSD 6.0-p3, KDE 3.5

PC-BSD выглядит зрелым проектом, за плечами которого целая серия дистрибутивов ветки 0.x. Релизы выходят достаточно регулярно, стараясь идти в ногу со временем, и нынешний уже основан на *FreeBSD 6.0*, что позволит вам рассчитывать на все преимущества последней версии. Удобный инсталлятор проведёт вас по всем этапам установки, снабжая процесс необходимыми пояснениями (правда, по-английски) и предоставляя там, где это нужно, возможность достаточно гибко влиять на параметры установки (например, при подготовке дискового раздела). В то же время, на каждый вопрос предлагается (как правило) разумный ответ, так что новичкам будет достаточно просто со всем соглашаться.

Перезагрузившись, вы окажетесь в KDE. Приложений немного, как и в *DesktopBSD* (см. ниже). Разве что можно скоротать пару часов за стандартными играми.

Отличительной особенностью дистрибутива является уникальная система управления пакетами – PBI. Вам достаточно скачать PBI-пакет программы с сайта **www.pbidir.com** и запустить установку двойным щелчком мыши. Аналогичный подход применяется и в настольных дистрибутивах Linux – вспомним хотя бы Click'N'Run Warehouse от Linspire. Набор пакетов в каталоге pbiDIR не всеобъемлющ, но обширен. Вы легко можете найти здесь *Mozilla Firefox* и *Thunderbird* (правда, версий 1.0.x), *Opera 7* и *8.50*, *OpenOffice.org 2.0*, подключаемые модули *Macromedia Flash*, *Java*, *Skype*, кодеки для различных мультимедиа-форматов, драйвера nVidia (но не ATI), слой двоичной совместимости с Linux (*Linux Binary Compat 1.0*) и Windows (*wine*, но не самой последней версии), *Scribus*, DesktopBSD Tools (!), игры, а также средства разработки и даже кое-какое серверное ПО (Webmin, Apache).. Если для нужного вам приложения пакет отсутствует, то путь один – ставьте коллекцию портов, и дальше – вручную.

Ну и если хочется иметь русскоязычный интерфейс – придется установить нужные пакеты (почему-то не **pbi**, а обычные **tgz**) со второго диска.



PC-BSD **можно поставить за одну только красоту, которой так мало осталось в нашем жестоком мире…**

РАБОЧЕМ СТОЛЕ

DesktopBSD

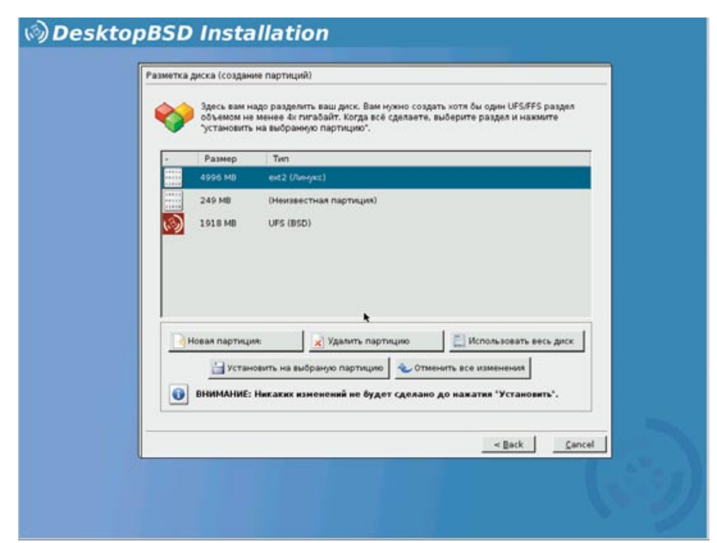
- **САЙТ ПРОЕКТА:** www.desktopbsd.net
- **ТЕКУЩАЯ ВЕРСИЯ:** 1.0RC3 (1CD для X86, 1DVD для AMD64)
- **ОСНОВАН НА:** FreeBSD 5.4-p8, KDE 3.4

Несмотря на возраст (как отмечается в FAQ, *DesktopBSD* примерно на год старше *PC BSD*), этот проект можно отнести к категории «развивающихся», в том плане, что на данный момент он выглядит скорее как любительский. Не очень информативный сайт, выпуск лишь трёх «кандидатов в релизы»… Тем не менее, к RC3 проект подошёл чуть больше чем за полгода, и уже сейчас эта версия выглядит обещающе. К тому же, помимо CD-версии для платформы x86, предлагается DVD-версия (правда, размером всего 930 Мб) для AMD64. Единственный минус – разработка всё ещё ведется на базе 5-й ветви *FreeBSD*.

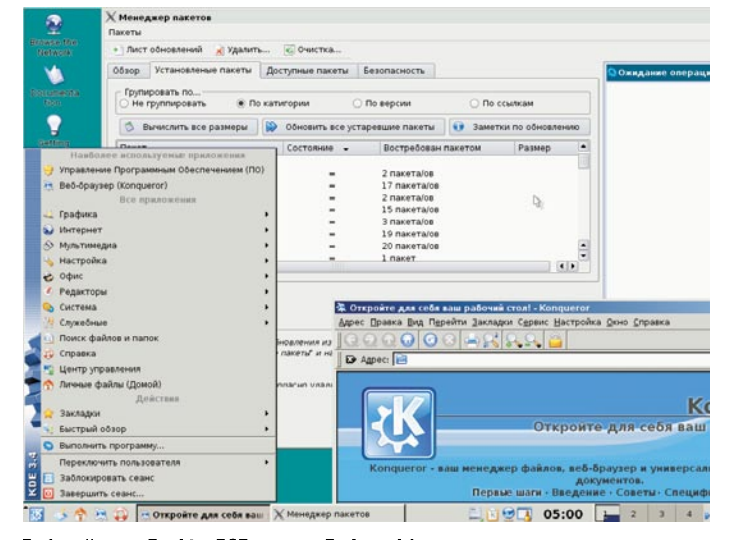
Удобный инсталлятор (поддерживается и русский язык) не вызовет никаких проблем – просто ответьте на некоторые вопросы, укажите раздел, на который вы собираетесь выполнить установку (создание подразделов будет выполнено автоматически), подождите, пока будут копироваться файлы… Вот, собственно, и всё – осталось создать пользователя, перезагрузиться, ответить ещё на пару вопросов по настройке графического окружения, и после прочтения небольшого вступления по работе с системой вы окажетесь в привычной атмосфере KDE.

Правда, особым богатством приложений дистрибутив не блещет. Как ни печально, но сразу после установки вы не найдёте ни офисного пакета, ни мультимедиа-приложений, ни даже игр (почти как свежеустановленная Windows) – лишь несколько текстовых редакторов, простенькие инструменты работы с графикой, ряд системных утилит.

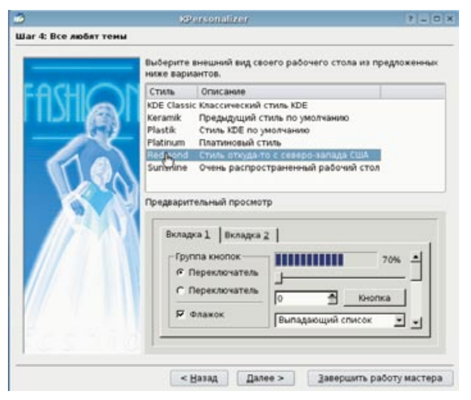
Изюминкой *DesktopBSD* является набор графических утилит *DesktopBSD Tools*, интегрирующихся с центром управления *KDE Control Center* и упрощающий выполнение повседневных задач, таких как настройка сетевого подключения, монтирование и размонтирование устройств, создание, редактирование и удаление учетных записей пользователей и так далее. Удачной находкой является также графическая оболочка к системе портов (своего рода менеджер пакетов), который делает установку нового ПО удобной и приятной. Особенно если у вас широкий канал в Интернет и бесплатный трафик…



При установке *DesktopBSD* главное – не поубивать нужное…



Рабочий стол *DesktopBSD* в стиле Redmond («стиль откуда-то с северо-запада США», как сказано в конфигураторе).



«Все любят темы»: постинсталляционная настройка *DesktopBSD*.

Заключение

Итак, вы увидели всё своими глазами. Установить один из рассмотренных дистрибутивов или самостоятельно добираться до желаемого результата, обзаведясь «официальными» дисками *FreeBSD* – ваше суверенное право. Кто-то скажет, что если вы боитесь консоли, то и *FreeBSD* вам ставить не стоит, и в какой-то степени будет прав. Но ведь и Linux когда-то начинался с одинокого ядра, на который бессонными ночами приходилось навешивать всё то, что необходимо для получения гордого звания «операционная система».

В любом случае, рассмотренные дистрибутивы можно считать хорошей отправной точкой для развёртывания на вашем рабочем столе системы *FreeBSD*. Многое ещё придётся доделывать вручную, но несколько часов вы, безусловно, сэкономите.



XEN: виртуальный роман

Виртуализуйте оборудование и удивите друзей своим мастерством. **Пол Хадсон** покажет, как справиться с установкой Xen.

Пускай у вас мощнейший компьютер – этакий Athlon FX-60, 4 ГБ ОЗУ, 1000 ГБ жесткого диска и Linux-система, и все-таки единственная ошибка при обновлении операционной системы может вывести его из строя. Произошло повреждение файла настроек, обнаружились ошибки в Glibc или пропажа модуля ядра – и ваша ОС рухнула: придется восстанавливать систему, используя резервную копию.

Но на грабли наступать вовсе не обязательно: клонируйте установленную ОС, запустите ее на виртуальной машине и выполните пробное обновление виртуально. Если не обошлось без ошибок, то обновление можно отменить, виртуальную машину отключить, а ваша система останется невредимой.

Это только один из способов применения виртуализации, на самом деле их гораздо больше. Виртуализация произвела революцию на рынке серверов, так как позволяет снизить объем неиспользованных ресурсов. К примеру, два сервера, допустим, с Apache и MySQL, загруженных на 25%, можно разместить на одном физическом сервере, организовав на нем две виртуальные машины – затраты на оборудование и обслуживание уменьшатся вдвое. Другой популярный пример использования виртуализации в бизнесе –создание кластеров и их использование для обеспечения дополнительной надежности, а также для отработки сложных сценариев на простых ПК.

Энтузиасты заинтересуются и просто шансом поиграть с новой технологией. Владельцам домашних компьютеров виртуализация предоставляет возможность одновременно запустить несколько ОС, например, Fedora Core 4 и Fedora Core 5 Test 2, а также SUSE 10.0, NetBSD и в некоторых случаях Windows. Круто!

С чего начать

Самое свежее решение по виртуализации – программа *Xen* – явилось из мира Open Source. Мы рассказывали о *Xen* в LXF 67 и предположили, что за ней будущее виртуализации. Так вот, это будущее настало!

Что особенного в данной технологии? Подход Xen сильно отличается от так называемой аппаратной виртуализации, применяемой в *VMware* и других укоренившихся продуктах. В VMware гостевая ОС остается в неведении о своем виртуальном статусе, а в *Xen* исходный код гостевой ОС должен быть немного подправлен, чтобы она понимала: ее место – под *Xen*. Это похоже на отношение симбиоза в природе: гостевая ОС знает о существовании *Xen* и реагирует на команды, а базовая ОС знает о своих гостях и получает от них информацию.

Новый подход окрестили «паравиртуализацией». На практике, паравиртуализация означает, что работа виртуальной машины (Virtual Machine, VM) не требует трудоемких операций, так как она осознает свой виртуальный статус и поэтому разделяет доступ ко всем ресурсам с остальными машинами без вмешательства *Xen*.

В результате скорость работы *Xen* VM всего лишь на 3–8% меньше, чем запуск системы на настоящем оборудовании, поэтому «почувствовать разницу» не удастся.

Принцип работы Xen

С тех пор, как мы рассказывали о *Xen* с десяток выпусков назад, она вошла в состав многих популярных дистрибутивов. SUSE, Fedora, Ubuntu, Gentoo и другие включили поддержку *Xen*, хотя качество поддержки зависит от дистрибутива. *Xen* негласно является одной из архитектур, под которые создается ядро Linux, наряду с x86, x86-64 и PowerPC.

Формально, когда вы устанавливаете *Xen* на компьютер, она становится главной операционной системой. Загрузка компьютера сопровождается загрузкой *Xen OS*, которая создает первоначальную виртуальную машину (известную как домен 0 или просто dom0), т.е. вашу базовую ОС. Все выглядит так, словно вы загружаете тот же Linux и затем X, как и до этого, но на самом деле вы используете виртуальную машину. Гостевая ОС dom0 является самой привилегированной, и поэтому имеет более высокий приоритет при доступе к оборудованию и вычислениях, чем другие гости. Этим гостям – обычно их называют непривилегированными доменами (domU) – требуется собственная порция оперативной памяти, а также постоянное место на диске для хранения своих данных.

Установка ядра

Хотя большинство дистрибутивов имеют поддержку *Xen*, мы потренируемся в сборке из исходных текстов, для гарантии, что вы сможете повторить процедуру установки с любым дистрибутивом. В данном руководстве мы будем использовать дистрибутив Fedora Core 4.

Прежде, чем мы начнем, необходимо сделать замечание: *Xen* должна работать с реальным оборудованием. Вы не сможете установить ее внутри *VMware* или с использованием других средств виртуализации.

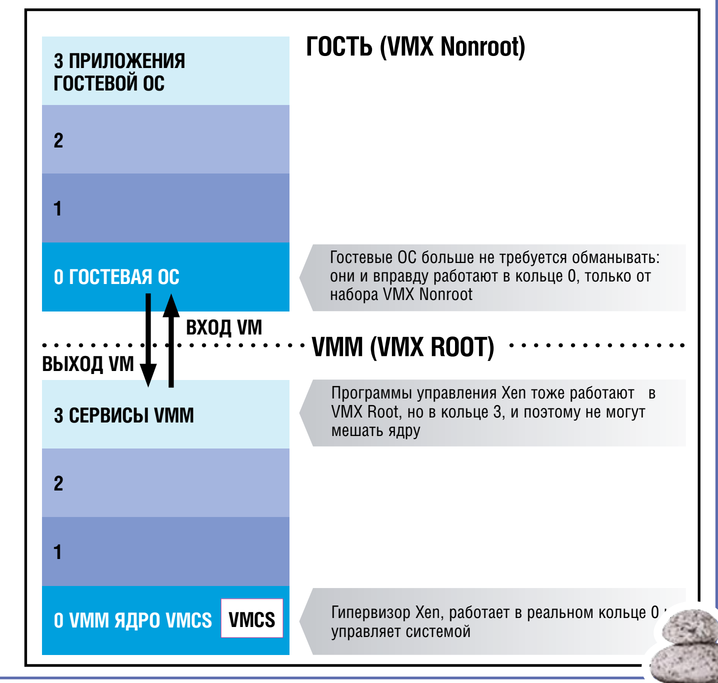
АППАРАТНАЯ ВИРТУАЛИЗАЦИЯ

Паравиртуализация устраняет падение производительности, хотя придется подправить исходный код гостевой ОС. Но это относится не ко всем архитектурам CPU. Архитектура Intel (x86–386, Pentium 4) а также клоны AMD и многие другие процессоры полагаются на иерархическую систему колец для назначения привилегий. Внутри процессора каждая программа (включая ОС) работает в кольце 0, 1, 2 или 3. В кольце 0 программы имеют полный контроль над машиной, а программы кольца 3 наиболее ограничены. На большинстве компьютеров ОС работает в кольце 0, а пользовательские программы в кольце 3, то есть даже самые агрессивные пользователи физически не могут отпихнуть ОС от ресурсов.

Проблема, связанная с кольцевой архитектурой, заключается в следующем: операционная система воображает, что будет работать в кольце 0. Но виртуальную ОС в кольце 0 запустить нельзя: это угрожает безопасности базовой ОС. Поэтому виртуальная ОС запускается в кольце с меньшим уровнем привилегий. Когда ОС проверяет, в каком кольце она работает, и получает ответ – не 0, ее работа аварийно завершается. Данная проблема обычно решается т.н. двоичной корректировкой: VM отслеживает запросы гостевой ОС на номер кольца и подменяет ответ. Двоичная корректировка – довольно трудоемкий процесс, поэтому традиционные техники виртуализации требуют мощного оборудования.

Паравиртуализация снимает проблему: ОС больше не претендует на работу в кольце 0, значит, никакая двоичная корректировка не нужна. Но что делать, если ОС невозможно подправить? Intel решила задачу с помощью технологии Vanderpool (сейчас просто VT), а AMD использует технологию Pacifica, которая заодно виртуализует контроллер памяти, тоже встроенный в чипы AMD64. Новые чипы работают с двумя наборами привилегированных колец: VMX Root и VMX Nonroot. Оба набора имеют кольца с 0 по 3, но только в наборе VMX Root кольцо 0 имеет полный контроль над компьютером. Набор VMX Nonroot виртуализуется чипом: когда немодифицированная ОС проверяет, в каком кольце она работает, то получает ответ `0`, но не может угадать, какому из двух наборов это кольцо принадлежит.

VT и Pacifica все равно требуют программного обеспечения для поддержки виртуализации гостевых ОС и взаимодействия с пользователями. *Xen* поддерживает обе технологии, поэтому, если у вас имеется подходящее оборудование, вы можете запускать немодифицированные ОС (в том числе Windows XP) без потерь производительности.



КАК УСТАНОВИТЬ XEN

Хватит болтать – установим-ка *Xen* и запустим нашу основную ОС как виртуальную машину.

1. Устранение зависимостей

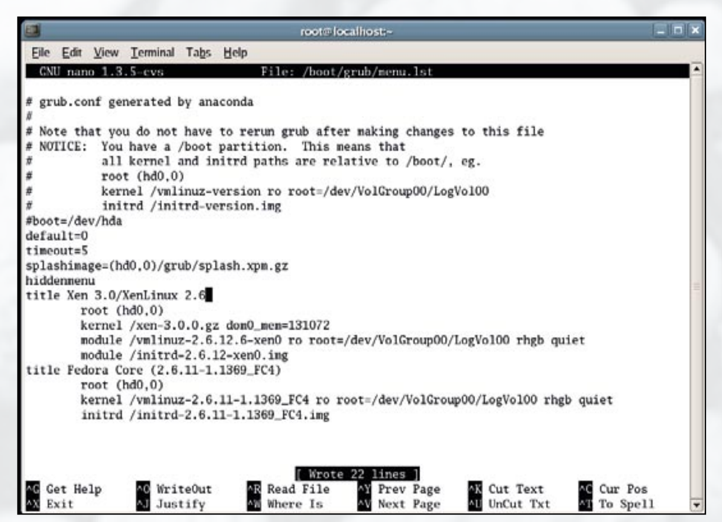
У *Xen* достаточно много зависимостей, и если вы хотите насладиться виртуализацией по максимуму, то их необходимо устранить. Вам потребуются полный набор для компиляции программ (*GCC*, *make* и др.), VNC (серверная и клиентская части), *libvncserver*, библиотека *python-twisted*, а также программа *brctl*, Curl (включая библиотеки разработки) и *SDL* (также включая библиотеки разработки).

Если вы устанавливаете Fedora Core 4, выберите Custom Setup. Отключите брандмауэр и SELinux (они могут вызывать проблемы), затем выберите следующие пакеты из предложенного списка:

- Из категории `X Window System` выберите **xorg-x11-Xnest**.
- Пометьте Development Tools (список выбранных пакетов по умолчанию нас устраивает)
- Из категории System Tools выберите *VNC*.
- Из Network Servers выберите *vncserver*.

Когда установка закончится, проведите первоначальную настройку системы. Зайдите в систему и загрузите Gnome. Если вы используете не FC4, менеджер пакетов вашего дистрибутива наверняка может предложить эквиваленты вышеназванных программ. Установка *Xnest* необязательна, но без остальных пакетов не обойтись.





Текстовый редактор *Nano* в Fedora не хуже других. Не забудьте убедиться, что в нем отключена опция автоматического переноса длинных строк.

<<

Теперь нам необходимо получить *libvncserver*, следовательно нам на сайт **http://libvncserver.sourceforge.net**. На момент написания статьи там была доступна для скачивания «предварительная версия» в виде файла **LibVNCServer-0.8pre.tar.gz**. Откройте терминал, перейдите в директорию, в которую скопировали его, и наберите

```
tar xpvfz LibVNCServer-0.8pre.tar.gz
cd LibVNCServer-0.8pre
./configure
Make
su -
make install
```

Если у вас Fedora Core, то остальные пакеты устанавливаются командой

```
yum install python-twisted bridge-utils curl SDL-devel
```

При использовании другого дистрибутива, менеджер пакетов может предложить их аналоги. Если вы не можете найти пакет **bridge-utils**, то поищите тот, который предоставляет команду *brctl*.

Теперь пора установить Xen. С сайта **www.cl.cam.ac.uk/Research/SRG/netos/xen/downloads** скачайте файл **xen-3.0.0-src.tgz**. Если доступна более новая версия, можете попробовать ее, но тогда мы не гарантируем, что дальнейшие инструкции будут работать. Наша версия находится на диске к журналу в директории **Magazine/Xen**.

2. Сборка ядра Xen

Получив файл с Xen, отправляйтесь обратно в терминал и выполните следующие команды:

```
tar xvpfz xen-3.0.0-src.tgz
cd xen-unstable
```

На диске в директории **Magazine/Xen** вы найдете специальную версию ядра Linux. Если вы скопируете ядро в свежесозданную директорию **xen-unstable**, *Xen* использует его для сборки своего образа. Если вы не скопируете наше ядро, то *Xen* скачает его из сети, да и вы сами можете поступить так же! Итак, ядро скопировано; перейдите в директорию **xen-unstable** и запустите

```
make world
```

Вне зависимости от способа установки ядра Linux, ядро будет извлечено, модифицировано для работы с *Xen*, а затем собрано дважды: сначала как xen0 (для dom0), а затем как xenU (для domU). Компиляция – процесс долгий, поэтому у вас будет достаточно времени на какие-нибудь другие дела.

После завершения компиляции можно двигаться дальше:

```
su
make install
```

Новые ядра *Xen* скопируются в директорию **/boot**, а файлы конфигурации – в вашу систему. Чтобы компьютер загружался с нашим ядром, необходимо создать образ начального RAM-диска, используя следующую команду

```
mkinitrd /boot/initrd-2.6.12-xen0.img 2.6.12.6-xen0
```

Первый параметр – путь, куда надо сохранить образ диска, а второй – имя директории с модулями в **/lib/modules**.

3. Конфигурация загрузчика

Чтобы добавить ядро xen0 в загрузчик *Grub*, откройте в текстовом редакторе файл **/boot/grub/menu.lst**. По умолчанию в Fedora он выглядит следующим образом:

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora Core (2.6.11-1.1369_FC4)
root (hd0,0)
kernel /vmlinuz-2.6.11-1.1369_FC4 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
initrd /initrd-2.6.11-1.1369_FC4.img
```

Необходимо привести его к следующему виду

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Xen 3.0/XenLinux 2.6
root (hd0,0)
kernel /xen-3.0.0.gz dom0_mem=131072
module /vmlinuz-2.6.12.6-xen0 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
module /initrd-2.6.12-xen0.img
title Fedora Core (2.6.11-1.1369_FC4)
root (hd0,0)
kernel /vmlinuz-2.6.11-1.1369_FC4 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
initrd /initrd-2.6.11-1.1369_FC4.img
```

Заметим, что теперь мы загружаем ядро *Xen*, а не Linux. Мы передаем имя Linux- ядра в *Xen*, и она запускает его в качестве гостевой ОС виртуальной машины dom0. Параметр **dom0_mem** определяет, сколько ОЗУ мы хотим выделить **dom0**. В нашем случае это 128 МБ. Можно указать любое другое значение, но не забудьте умножить количество мегабайт, которое вы хотите выделить виртуальной машине, на 1024.

Сохраните файл, перегрузитесь, и теперь по умолчанию система грузит ядро *Xen*, которое может не содержать нужных вам настроек – тогда придется отредактировать его конфигурацию или пересобрать нужные вам драйверы.

Теперь войдите в систему, запустите консоль и наберите **uname -a**. `xen0` в строке вывода результата команды сигнализирует о том, что вы теперь в домене 0. Переключитесь в режим root с помощью команды **su –** и запустите **xend start**, а потом **xm list**. Последняя команда выводит список всех работающих в данный момент виртуальных машин. В этом списке вы видите dom0, которой выделено 128 МБ ОЗУ. Поздравляем! Ваша система заработала в *Xen*!

Совет Xen: Отсоединение

Не забывайте, что все ваши виртуальные машины работают в фоновом режиме. Поэтому если вы закроете терминал, с которого работали с доменом, он не остановит свою работу и даже не приостановится. Чтобы заново подсоединится к домену, необходимо использовать следующую команду: xm console Linux1 (вместо Linux1 поставьте имя вашего домена). Для отсоединения от домена нажмите Ctrl +]

Xen на предприятии

Вручение *Xen* администратору равносильно выдаче автомата Калашникова неандертальцу: техника в руках дикаря приносит скорее вред, чем пользу, потому что требует специальных знаний и навыков. Крупная фирма не станет действовать «в лоб» и сокращать количество серверов, загруженных на 25%, просто распределив их на меньшем количестве серверов с загрузкой 90% и выбрав *Xen* в качестве средства виртуализации. Конечно, такая возможность есть, но переход потребует огромных затрат на планирование, тестирование и внедрение. На наш урок вам потребуется час или около того; а сколько понадобится времени, если вы работаете на ответственной системе? Или если бы вам пришлось дублировать точные конфигурации сервера, а то и – самое сложное – развертывать *Xen* на множестве серверов и добиваться их слаженной работы?

Сама по себе *Xen* – просто инструмент; в ее состав не входят другие средства управления, кроме рассмотренных здесь. Однако фирма *XenSource*, разработчик *Xen*, предусмотрела проприетарный программный продукт для управления, именуемый *XenOptimizer*. Он позволяет предприятиям развертывать и управлять *Xen* в вычислительных центрах, предоставляя консоль, с которой администратор может контролировать ресурсы, создавать и модифицировать виртуальные машины *Xen*, а также осуществлять динамическое распределение нагрузки на сервера, чтобы отвечать любым потребностям пользователей. Если вы намерены развертывать *Xen* на более чем 10 физических серверах, то *XenOptimizer* может сэкономить время и деньги и даже спасти положение.

www2.getxen.com/products/xenoptimizer.html.



Во время установки Fedora Core 4, выберите Custom Setup, т.к. нам понадобятся дополнительные пакеты.

Создание виртуальной гостевой ОС

Работали мы, работали… и что вышло? Ваша система работает так же, как и до этого, верно? Нет, не совсем: теперь можно создавать непривилегированных гостей domU. Приступим…

1. Создание образов диска

Переключитесь в суперпользователя и запустите **df -h**, чтобы проверить, сколько свободного места в системе. Вам понадобится 4 ГБ свободной памяти, однако примите во внимание объем текущей инсталляции Linux. Чтобы запустить гостя domU, необходимо сначала создать образ для его корневого раздела, а также образ для раздела подкачки. Следующие две команды создадут образ корневого раздела. Выполнять их надо из директории **/root**:

```
dd if=/dev/zero of=vmdisk bs=1k seek=4096k count=1
mkfs.ext3 vmdisk
```

За запуском второй команды последует предупреждение, что файл не является специальным блочным устройством; для продолжения, просто нажмите [y]. Установка параметра **seek** как 4096k дает нам образ диска размером в 4 ГБ; вы можете менять размер по своему усмотрению: 1024 соответствует 1 ГБ, 2048 – 2 ГБ, 8192 – 8 ГБ и т.д. В данный момент диск отформатирован как ext3, но он совершенно пуст, поэтому необходимо сделать на нем клон нашей системы.

Следующие команды монтируют образ как loopback-устройство, а затем копируют на него вашу ОС. Если у вас много чего установлено или вы хотите скопировать только некоторые части системы, тогда придется быть более точным.

Прежде чем продолжить, ответьте на следующие вопросы:

- Вы проверили, какой объем занимает ваша установленная система с помощью **df -h**?
- Вы создали образ диска необходимого объема?
- Вы создали файл **vmdisk** в каталоге **/root**?

Если ответом на все три вопроса было «да», тогда двигаемся дальше. Выполните следующие команды:

```
mkdir /mnt/vmdisk
mount –o loop vmdisk /mnt/vmdisk
cp –ax /{ bin,dev,etc,home,lib,sbin,usr,var} /mnt/vmdisk/
mkdir /mnt/vmdisk/{proc,sys,tmp,root}
chmod og+w /mnt/vmdisk/tmp
```

Команде *cp -ax* требуется ощутимое время для исполнения. Директории **proc**, **sys**, **tmp** и **root** копировать не нужно, они просто создаются командой **mkdir**. В некоторых дистрибутивах необходимо также скопировать каталог **/opt**, однако под Fedora этого не понадобится.

Последняя команда меняет права доступа к **/tmp**, разрешая программам записывать туда свои данные.

Гость dom0 имеет прямой доступ к оборудованию, в то время как гость domU будет использовать образ диска для хранения данных. Поэтому им необходимы разные настройки *fstab*. Откройте в текстовом редакторе файл **/mnt/vmdisk/etc/fstab**. В Fedora он будет выглядеть следующим образом:

```
# This file is edited by fstab-sync - see 'man fstab-sync' for details
/dev/VolGroup00/LogVol00 / ext3 defaults 1 1
LABEL=/boot /boot ext3 defaults 1 2
/dev/devpts /dev/pts devpts gid=5,mode=620 0 0
/dev/shm /dev/shm tmpfs defaults 0 0
/dev/proc /proc proc defaults 0 0
/dev/sys /sys sysfs defaults 0 0
/dev/VolGroup00/LogVol01 swap swap defaults 0 0
/dev/hdc /media/cdrom auto pamconsole,exec,noauto,managed 0 0
```

Как видите, Fedora использует логические тома (LVM) для управления жесткими дисками, о чем говорит запись **VolGroup00**. Однако ваш файл **fstab** может просто использовать **/dev/hda1** и **/dev/hda2**. В нашем случае необходимо отредактировать файл, чтобы он монтировал **hda1** и **hda2** в качестве корневого раздела и раздела подкачки соответственно. Исправленный файл должен выглядеть так:

```
/dev/hda1 / ext3 defaults 1 1
/dev/devpts /dev/pts devpts gid=5,mode=620 0 0
/dev/shm /dev/shm tmpfs defaults 0 0
/dev/proc /proc proc defaults 0 0
/dev/sys /sys sysfs defaults 0 0
/dev/hda2 swap swap defaults 0 0
```

Сохраните его и выполните **unmout /mnt/vmdisk**, чтобы размонтировать виртуальный образ. Это важно, т.к. *Xen* не может использовать файловые системы, которые уже смонтированы Linux. Теперь создадим виртуальный диск для раздела подкачки объемом 256 МБ, следующим образом:

```
dd if=/dev/zero of=vmswap bs=1k seek=256k count=1
mkswap vmswap
```

2. Создание файла настройки

Создание дисков для виртуальной машины закончено, осталось настроить файл конфигурации. Выполните команду:

```
cp /etc/xen/xmexample1 ~/xmlinux1
```

В результате файл с примером конфигурации *Xen* скопируется в текущую директорию. Откройте его в текстовом редакторе.

Необходимо сделать несколько изменений – найдите следующие строки и исправьте их на указанные значения:

- `kernel=` в **/boot/vmlinuz-2.6.12-xenU**.
- `memory=` в 128
- `name=` в Linux1
- `disk=` в ['file:/root/vmdisk,hda1,w', 'file:/root/vmswap,hda2,w'].

Если ваша версия ядра отличается от нашей, то подправьте файл соответствующим образом. Память измеряется в мегабайтах; мы работаем со 128 МБ ОЗУ и 256-МБ диском подкачки, однако если у вас система мощнее, можете определить собственные значения параметров. Имя Linux1 дано, чтобы отличить VM от других (к тому же его легко запомнить); у каждой VM должно быть уникальное имя.

Наконец, параметры диска определяют, что **/root/vmdisk** монтируется как **hda1** (корневой раздел гостя domU) с правами запись/чтение, а **/root/vmswap** монтируется как **hda2** (файл подкачки гостя domU), также с правами запись/чтение.

Теперь можно запустить новую VM!

```
xm create -c xmlinux1
```

Параметр **–c** позволяет определить наш файл конфигурации для новой виртуальной машины. Если появляется ошибка 'Error connecting to xend: Connection refused', значит, вы забыли выполнить **xend start** от имени root на предыдущем шаге. Также может появиться ошибка 'File /root/vmdisk is loopback-mounted through /dev/loop0'. Она говорит о том, что вы не отмонтировали образ виртуального диска.

Ошибка 'Error creating domain: VM name 'Linux1' already in use` говорит сама за себя. Обычно она появляется после другой ошибки: *Xen* создает VM, пытается запустить ее и терпит неудачу, но созданную VM при этом не уничтожает. Выполнив **xm list**, вы увидите, что Linux1 находится в состоянии **p** (приостановлена). Уничтожьте ее с помощью **xm destroy Linux1** и создайте заново.

Если не произошло ни одной ошибки (считайте себя счастливчиком!), вы увидите, как начнет загружаться новая машина. В конце концов появится приглашение войти в систему. Используйте ваш логин и пароль, т.к. мы скопировали **/etc/passwd** на виртуальный диск.

>>

Совет Xen: о миграции машин

Рассмотрим наиболее мощную и впечатляющую возможность *Xen*: миграцию виртуальных машин, позволяющую запускать VM одного компьютера на другом и при этом не тратить и десятой доли секунды на системный простой.

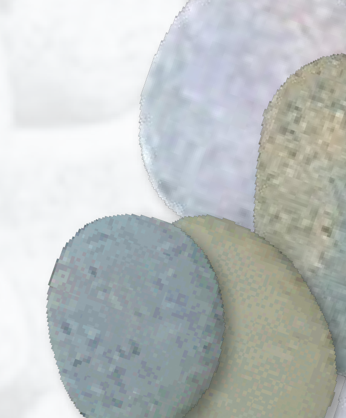
Все потому, что миграция VM протекает в форме копирования образа памяти системы на другой компьютер; файловая же система должна находиться в месте, доступном обоим компьютерам, то есть у вас должно быть NAS- или SAN-устройство. Если у вас под рукой оказалось 2 сконфигурированных компьютера с *Xen*, а также NAS, на котором хранятся файлы VM, то можно сделать следующее:

```
xm migrate --live Linux1 ДругойСервер.ВашаСеть.com
```

Виртуальная машина скопируется на другой компьютер, затем произойдет ее синхронизация с исходной, и после этого *Xen* осуществит переключение. Без использования флага --live произойдет приостановка VM, копирование ее на другой компьютер и перезапуск, простой системы увеличится.

Трудности перевода

1) NAS (Network Attached Storage) – устройство, функционирующее на базе протокола TCP/IP и применяющееся в качестве централизованного файлового сервера.
2) SAN (Storage Area Network) – выделенная сеть, предназначенная для высокоскоростного соединения устройств хранения данных с серверами, а также серверов между собой.



РАБОТА С XEN

В данный момент у нас работает Linux на Linux, хотя и только в командной строке. Посмотрим, как можно переключиться в графический режим, а также как управлять и работать с *Xen*.

УПРАВЛЕНИЕ CPU

В отличие от других техник виртуализации, при паравиртуализации, примененной в *Xen*, гостевая ОС знает о своем виртуальном статусе.

На практике это означает, что мы можем посылать гостевой ОС команды и получать от нее ответ. Например, откройте второй терминал (первый оставьте работать с гостевой ОС dom0), переключитесь в суперпользователя с помощью **su -** и выполните команду

```
xm shutdown Linux1
```

Нажав [Enter], вы увидите, что на первом терминале появляются стандартные сообщения завершения работы Linux. Другими словами, *Xen* не выдернула VM из розетки: VM сама себя остановила и вернула контроль родительской ОС.

Этим команда *shutdown* отличается от команды destroy, которая как раз убивает VM. Уничтожение VM с помощью *destroy* может повредить диск, поэтому ее не рекомендуется пускать в ход, пока ваша VM не будет приостановлена.

Вернувшись в командную строку главной виртуальной машины и перейдя в режим суперпользователя, откройте в текстовом редакторе файл **xmlinux1**, найдите строку с параметром vcpus и установите его значение равным 4. Если перед параметром стоит знак комментария (символ #), удалите его. Теперь перезапустите машину с помощью

```
xm create –c Linux1
```

и войдите под своей учетной записью. В выводе команды **cat /proc/cpuinfo** вы увидите, что в наличии имеется 4 CPU. Физически у нас всего лишь 1 процессор на материнской плате, но *Xen* это не волнует: она виртуализует столько CPU, сколько потребуется. Если у вас в системе физических процессоров несколько, можно изменить параметр `cpus` в файле **xmlinux1** и указать количество процессоров, на которых должна работать VM.

```
root@localhost:~
File  Edit  View  Terminal  Tabs  Help
root@localhost ~]# xm vcpu-set Linux1 2
root@localhost ~]# xm list
ame                                    ID Mem(MiB) VCPUs State   Time(s)
omain-0                                 0      128     1 r-----     48.3
inux1                                   1      128     2 ------     21.8
root@localhost ~]# xm vcpu-set Linux1 4
root@localhost ~]# xm list
ame                                    ID Mem(MiB) VCPUs State   Time(s)
omain-0                                 0      128     1 r-----     49.2
inux1                                   1      128     4 -b----     22.1
root@localhost ~]# xm vcpu-set Linux1 8
root@localhost ~]# xm list
ame                                    ID Mem(MiB) VCPUs State   Time(s)
omain-0                                 0      128     1 r-----     49.8
inux1                                   1      128     8 -b----     22.6
root@localhost ~]# xm vcpu-set Linux1 16
root@localhost ~]# xm list
ame                                    ID Mem(MiB) VCPUs State   Time(s)
omain-0                                 0      128     1 r-----     50.4
inux1                                   1      128     8 -b----     22.6
root@localhost ~]#
```

Командой *xm vcpu-set* можно динамически менять число виртуальных ЦПУ в домене, но если это число превышает значение vcpus, установленное в файле конфигурации гостевой ОС, команда выполнена не будет.

УПРАВЛЕНИЕ ОЗУ

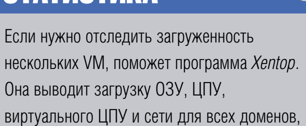
С процессорами мы научились управляться, но как насчет ОЗУ? Наберите **top** внутри гостевой машины domU. В левом верхнем углу вы увидите строку, похожую на 'Mem: 134900k total'. Это объем ОЗУ, который мы выделили машине. Теперь с другого терминала (вашей главной машины) запустите от имени суперпользователя **xm list**. Вы увидите, что домен Linux1 работает и ему действительно выделено 128 МБ ОЗУ. Давайте поменяем объем RAM! Выполните следующую команду:

```
xm mem-set Linux1 64
```

Мы уменьшили объем доступной Linux1 памяти до 64 МБ.Снова запустите команду *top* в гостевой машине domU. Вы увидите 'Mem: 65536k', то есть то, что нам нужно. Посмотрим, как вы сделаете это в *VMware Workstation*…

Команда *mem-set* позволяет изменять

СОВЕТ XEN: СТАТИСТИКА



Если нужно отследить загруженность нескольких VM, поможет программа *Xentop*. Она выводит загрузку ОЗУ, ЦПУ, виртуального ЦПУ и сети для всех доменов, все на одном экране.

БУДЬТЕ ГРАФИЧНЕЕ

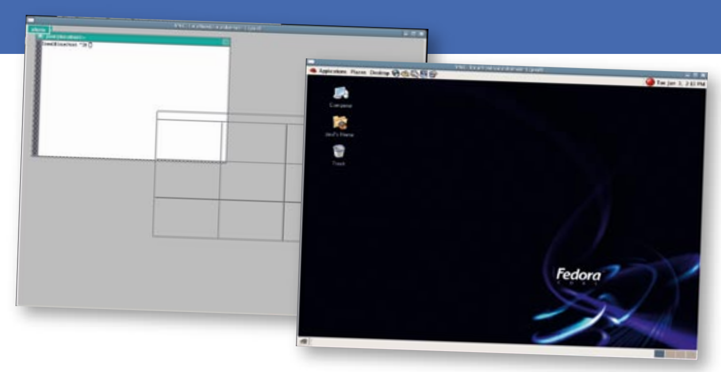
Запуск Gnome

До текущего момента гость domU работал только в текстовом режиме. Ничего удивительного: у гостей domU нет доступа к видеокарте, поэтому им уготована судьба вечно жить на командной строке. Однако расстраиваться не стоит, так как у *X*-сервера есть возможность работать по сети – в нашем случае, виртуальной сети. Можно запустить *VNC* (или другую программу по вашему усмотрению) в домене domU, который не имеет доступа к оборудованию, а работать с ним в dom0, где есть доступ к видеокарте.

Приступим! В терминале domU запустите **vncpasswd**, для установки пароля *VNC*; проверьте IP адрес системы с помощью **/sbin/ifconfig**. Наконец, наберите **vncserver** (запуск VNC-сервера).

Вывод команды *vncserver* очень важен, т.к. позволяет узнать номер дисплея, которого ожидает сервер – либо 0, либо 1; с увеличением числа *VNC*-серверов, этот номер увеличивается. Просмотрите вывод на наличие строки вида 'New 'ИмяМашины@ИмяДомена:1 (ИмяПользователя)' ' – указанное число и есть то, что мы ищем.

Теперь в машине dom0 запустите **vncviewer**. Появится приглашение ввести имя *VNC* сервера: введите IP адрес домена гостя domU, двоеточие, а затем номер дисплея, т.е. 192.168.133.63:1. Нажмите OK, введите пароль, и перед вами появится удаленный рабочий стол с простеньким оконным менеджером вроде *TWM*.Закройте окно с удаленным рабочим столом, перейдите в терминал domU и запустите **killall Xvnc**, чтобы остановить *VNC*-сервер. Находясь в этом же терминале, откройте в текстовом редакторе файл **~/.vnc/xstartup**. Перейдите в конец файла. Последняя строка будет выглядеть примерно так: **twm &**, что означает запуск оконного менеджера *TWM*. Измените ее на **gnome-session &**, сохраните файл и выйдите. Далее перезапустите *VNC* сервер командой **vncserver**. Вернитесь к машине dom0, запустите **vncviewer** и подсоединитесь к серверу. На этот раз вы окажетесь в Gnome! Правда, по умолчанию *VNC* использует 64 цветную палитру, нормальную для *TWM*, но совсем не приемлемую для работы с Gnome. Чтобы это исправить, надо в окне ввода IP адреса нажать на кнопку Options и выбрать значение Full для количества цветов. Не беспокойтесь о пропускной способности: *VNC* довольно быстро работает через сеть. А еще быстрее работает на машинах, находящихся в одной сети.



По умолчанию, оконный менеджер *VNC* – это *TWM* (слева)**, страшноватый на вид. Отредактируйте файл xstartup, и получите Gnome** (справа)**..**

Совет Xen: автозапуск

Если у вас установлено несколько виртуальных машин и вы хотите, чтобы они все запускались при загрузке системы, то скопируйте их файлы настройки (можно также создать символьные ссылки на них) в директорию **/etc/xen/auto** и обеспечьте выполнение скрипта *xendomains*. В Fedora добавить скрипт можно с помощью команды

```
chkconfig --level 345 xendomains on
```

В некоторых дистрибутивах это можно сделать с помощью программ, имеющих графический интерфейс. Аналогично добавляется скрипт *xend*.

размер ОЗУ, доступный виртуальной машине. Максимальный объем выделенной ОЗУ не может превышать значения, определенного в файле настройки. Т.е. вы можете создать машину с 256 МБ ОЗУ, а затем с помощью *mem-set* ограничить доступный объем до 8–16 МБ (в этом случае ядро начнет убивать процессы, чтобы высвободить память). Можно увеличить объем до 128 МБ или даже установить начальное значение 256 МБ, однако попытки выйти за пределы 256 МБ будут проигнорированы. Если хотите достичь большей гибкости, рекомендуем создавать виртуальные машины с большими начальными значениями ОЗУ (для этого измените значение `memory =` в файле **xmlinux1**), а затем уменьшать ОЗУ до реально имеющихся. Если же объем ОЗУ потребуется увеличить, у вас будет запас для маневра.

Сохранение и перезагрузка

Теперь рассмотрим возможности поинтереснее: т.к. ядро *Xen* управляет гостями Linux, можно сохранить состояние выбранной VM перед ее отключением! По существу, происходит создание дампа оперативной памяти VM. Его размер будет равен объему ОЗУ, выделенного виртуальной машине. Если захотите вернуться в сохраненное состояние, то *Xen* просто скопирует дамп обратно в ОЗУ. Не меняйте расположение файлов **vmdisk** и **vmswap** после сохранения состояния виртуальной машины. Сохранение и восстановление виртуальных машин можно осуществлять следующими двумя командами:

```
xm save Linux1 Linux1.state
xm restore Linux1.state
```

Первая команда сохраняет машину Linux1 в файл **Linux1.state**, затем происходит ее выключение. Вторая команда восстанавливает машину (если файл **Linux1.state** не находится в том же каталоге, что и вы, в команде придется указать его полный путь).

Одна из особенностей *Xen* – ее виртуальные машины работают на уровне, отличном от уровня базовой машины (домен 0). После создания и запуска виртуальной машины с помощью **xm create**, вы сможете войти в нее. Однако если вы закроете терминальное окно, VM будет продолжать работу в фоновом режиме, как если бы ничего не случилось. Ваше соединение с ней напоминает удаленный терминал: вы свободно подключаетесь и отключаетесь, а она невозмутимо работает.

Вместо того, чтобы просто закрыть терминал, попробуйте нажать Ctrl +] (правая квадратная скобка) для отсоединения от консоли VM и возврата в главную машину. Хотя VM работает, к ее выводу вы подсоединиться не можете. Чтобы вернуться в VM, наберите команду **xm console Linux1**. Возможно, потребуется нажать Enter еще раз, чтобы появилось приглашение командной строки. Консоль VM – не то же самое, что настоящий удаленный терминал, потому что, запустив в двух терминалах

```
xm console Linux1
```

вы окажетесь в одной сессии консоли (проверьте: запустите в одном терминале **uname –a**, затем переключитесь в другой и нажмите стрелку вверх, чтобы увидеть последнюю запущенную команду; вы увидите, что это *uname*).

Если хотите использовать настоящий удаленный терминал, попробуйте *SSH*. Выполните **/sbin/ifconfig** в гостевом домене xenU, а затем из xen0 (ваша главная машина) подключитесь к xenU с помощью *SSH*. LXF

Совет Xen: ускоряем файловые системы

В данном руководстве виртуальные машины использовали петлевые файловые системы: то, что они воспринимали как **hda1** и **hda2**, в действительности было файлами на жестком диске. Максимальная производительность достигается на настоящих жестких дисках. Их также рекомендуется использовать, если вам требуется более 8 файловых систем, т.к. по умолчанию в Linux предусмотрено одновременное подключение не более 8 loopback-устройств. Альтернатива – использование логических томов для виртуальных машин, что позволяет динамически менять размер виртуальных дисков и создавать снимки системы.

X сервер

Казалось бы, с установкой графического режима мы исчерпали все возможности *Xen* по работе с графикой. А вот и нет! Используя встроенные возможности *X* по работе с сетью и программу *Xnest* (которую, надеюсь, вы не забыли установить), мы можем запустить *X*-сервер внутри другого *X*-сервера без помощи VNC! Более того, посредством XDMCP *X*-сервер может сканировать сеть и обнаруживать все машины, разрешающие удаленное соединение.

Находясь в домене domU, откройте в тестовом редакторе файл **/etc/X11/gdm/gdm.conf**. Найдите строку [xdmcp] и измените 'Enable = false' на 'Enable = true'. В domU от имени суперпользователя наберите

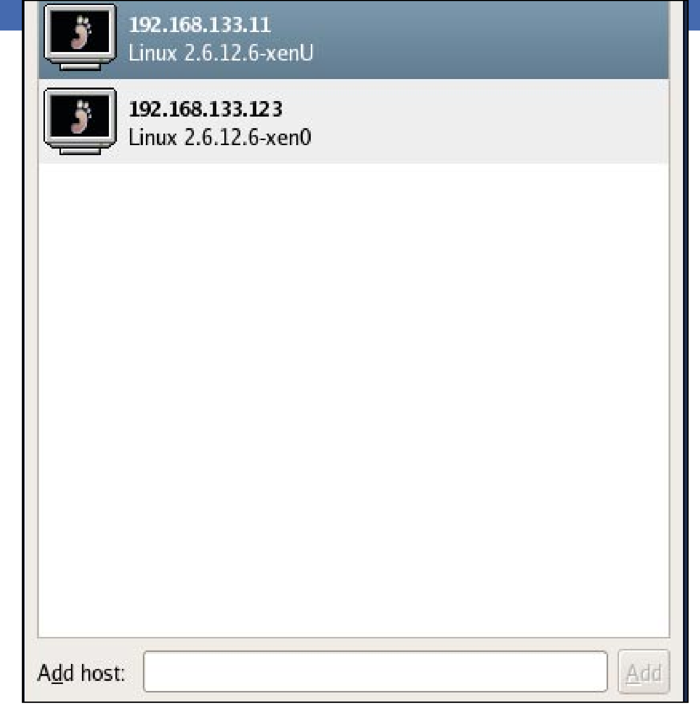
```
xm shutdown Linux1
```

чтобы завершить работу гостя Linux1. Затем откройте файл **xmlinux1** и найдите параметр Extra. Он отвечает за уровень запуска гостевой ОС – установите его равным 5 для активации *X*. Теперь запустите

```
xm create –c Linux1
```

чтобы создать домен.

Оставаясь суперпользователем в dom0, наберите **gdmsetup**, чтобы сконфигурировать менеджер входа в систему (в Fedora это *GDM*). В появившемся окне на закладке XDMCP выберите Enable XDMCP. Проверьте, что также выбран пункт Honour Indirect Requests и закройте окно. Наберите `exit`, чтобы стать обычным пользователем, а затем **gdmXnestchooser**. В результате появится окно и произойдет сканирование сети на наличие хостов с активированным XDMCP, к которым можно подключиться. Если все будет удачно, то вы увидите в списке машины dom0 и domU. Выберите для подключения машину domU, и готово!



С помощью *gdmXnestchooser* вы можете выбрать домен, к которому хотите подключиться.

К В@ШИМ УСЛУГ@М

Вам никогда не казалось, что ваша машина с Linux могла бы делать для вас больше, чем целый день запускать *OpenOffice.org* да «гонять» *Frozen Bubble*? Задайте вашему ПК настоящую работу, настроив его как суперсервер Интернета.



Многие считают, что Интернет существует лишь для того, чтобы пассивно пользоваться его благами. Многие тратят больше времени на просмотр сайтов, чем на их разработку. Многие даже не представляют, что могут настроить собственный BitTorrent. Но вы же не принадлежите к этим «многим»? Вы не нуждаетесь в напоминании, что ваш компьютер с Linux содержит самые мощные и популярные серверные программы на свете. Немного поизучав их, вы сможете превратить вашу настольную машину во «всегда к вашим услугам» суперкомпьютер, открытый всему миру. Он сможет поделиться вашими файлами, представить ваши web-страницы, стать чат-сервером, обслужить вашу радиостанцию, и многое другое – и ни одна из этих задач не потребует столько оперативной памяти, чтобы выделить для нее вспомогательный ПК.

Превратить ваш ПК в узел сети Интернет не сложно, но зато очень здорово, и что важнее всего – совершенно бесплатно. Читайте дальше.



НА ДИСКЕ

АДМИНИСТРАТОРЫ СЕРВЕРА

Эту статью подготовили для вас Пол Хадсон (Paul Hudson), Грэм Моррисон (Graham Morrison), Майк Сондерс (Mike Saunders), Ник Вейч (Nick Veitch).

«ВАШ LINUX-КОМПЬЮТЕР СОДЕРЖИТ САМЫЕ МОЩНЫЕ СЕРВЕРНЫЕ ПРОГРАММЫ НА СВЕТЕ.»

Публикуем файлы в BitTorrent

BitTorrent – замечательный способ поделиться файлами. Не желают ли дамы и господа запустить собственный сервер?

Не путайте с нелегальным распространением файлов: на самом деле *BitTorrent* – всего лишь невероятно эффективный протокол и замечательно реализованная идея, которая заслуживает большего, чем ассоциации с мрачным сетевым «подпольем».

BitTorrent эксплуатирует старую добрую модель клиент-сервер, когда те, кто скачивает файл, предоставляют секции, или «блоки», которые они уже скачали, другим участникам сети. Можно сравнить это с пирамидой из бокалов с шампанским. Шампанское льётся только в верхний бокал, и по мере заполнения перетекает в нижние. Только в случае с *BitTorrent* можно начинать делиться «шампанским» ещё до того, как ваш «бокал» наполнится.

Чтобы скачать поток, пользователь сперва должен получить копию специального Torrent-файла (с суффиксом .torrent). Он создается на основе исходного файла и содержит хэш для каждого отдельного блока, который можно скачать и предоставить остальным. Хэш гарантирует, что каждый фрагмент файла, скачанный пользователем, будет в точности совпадать с оригиналом.

Наиболее творчески реализовано распределение клиентами тех частей, которыми они уже располагают. Это зависит от трекера (tracker), интернет-адрес которого вложен в Torrent-файл. Трекер выполняет функции дирижёра. Он отслеживает, кто что скачал, и всякий раз при получении нового запроса на ту или иную закачку он гарантирует, что новый клиент получит кусочки, которых еще нет у других участников сети.

Вот что вы можете сделать на вашем ПК: запустить собственный трекер, так что вы не просто будете разделять потоки – вы будете их создавать.

Строим трекер

Наиболее популярный трекер называется *XBNBT*, первоначально его разработали на языке Python и поставляли с оригинальным *BitTorrent*, а потом переписали на C++. Вам нужно скачать исходный код *XBNBT* или взять его с нашего диска. Распакуйте файл и перейдите в каталог **bnbt/src**. Мы будем собирать простейший трекер, так что просто наберите **make xbnbt**. При этом в текущем каталоге будет создан исполнимый файл. Очень важно ограничить трекер пользовательской учётной записью, так что не обязательно этот файл куда-то устанавливать.

Следующий этап – создание каталога **torrents**, где будут размещаться Torrent-файлы, которыми будет управлять сервер. Его нужно создать внутри каталога, из которого будет запускаться исполняемый файл. Наконец, перед запуском сервера необходимо отредактировать замысловатый конфигурационный файл. Откройте **bnbt.cfg** и измените следующие строки (**torrents** указывает на каталог, который мы только что создали):

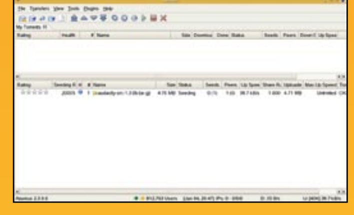
```
allowed_dir = torrents
bnbt_allw_torrent_downloads = 1
bnbt_file_dir = torrents
bnbt_parse_on_upload = 1
```

Теперь можно запускать сервер. Выполните команду **./xbnbt** в текущем каталоге. Вы должны получить на выходе информацию, в случае удачи содержащую фразу «Server Info – Start», которая означает, что трекер запустился. Значит, пора обратиться к трекеру, введя в строке браузера **http://localhost:6969** (замените «localhost» URL-адресом вашей машины). До создания новой учётной записи больше ничего делать не надо. Просто припишите к введённому URL **/users.html**, и откроется страница User Info. Создайте учётную запись администратора и соответствующий пароль, и отметьте все «флажки», чтобы получить все привилегии.

Создать Torrent-файл из файла, который вы хотите предоставить в общий доступ, могут многие популярные клиенты;



Сперва создайте трекер с помощью *XBNBT*.



***Azureus* умеет и создавать собственные потоки, и подключаться к вашему серверу.**

самый простой в использовании, вероятно, *Azureus*. Выберите пункт Create Torrent в меню File. Параметр «Announce URL», который будет запрошен на первой странице – это адрес вашего сервера плюс **/announce** (используя «localhost» в качестве адреса сервера, получим **http://localhost:6969/announce**). Щёлкните на Next и выберите файл, которым вы решили поделиться. *Azureus* продолжит работу и создаст Torrent-файл. Можете сохранить его где хотите.

Поделитесь потоком

Torrent-файл, который вы только что создали, нужно загрузить на трекер. Для этого откройте страницу **http://localhost:6969/upload.html**. Щёлкните по кнопке запроса, укажите ваш Torrent-файл и щёлкните на Upload, чтобы скопировать его на сервер.

Последний этап – самый важный, и на первый взгляд, неочевидный. Ваш сервер теперь умеет отслеживать поток, но прежде чем кто-либо сможет скачать его, вам нужно породить (seed) копию. В *Azureus* откройте поток, только что переданный на сервер, и когда *Azureus* спросит вас о пункте назначения, убедитесь, что вы указали место, где сохранён исходный файл. *Azureus* сопоставит этот файл с Torrent-файлом, проверит, не требуется ли скачать что-либо ещё, и сразу же запустит поток. Чудеса! Вот теперь можете публиковать ссылку на Torrent.

Последнее замечание: ваш трекер будет управлять каждым соединением с другими сидерами (seeder) и личерами (leecher). Проверьте на главной странице – **http://localhost:6969** – текущий статус трекера, включая все доступные потоки и число сидеров и личеров, подсоединённых к нему.

ЗАЧЕМ МНЕ ЭТО?

- Чтобы распространять файлы, не перегружая ваш интернет-канал
- Чтобы отслеживать популярность файлов, настроив трекер
- Чтобы использовать RSS-каналы для размещения новых данных для ваших пользователей



ТЕРМИНОЛОГИЯ

Трекер (tracker) – сервер, управляющий распространением файла
Сидер (seeder) – тот, кто отдает фрагменты файла
Личер (leecher) – тот, кто скачивает фрагмент файла

ПОЛЕЗНЫЕ ССЫЛКИ

- **http://sourceforge.net/projects/xbnbt** – домашняя страница ***XBNBT***.
- **http://azureus.sourceforge.net** – домашняя страница ***Azureus***.
- **http://azureus.sourceforge.net –**
- **http://xmlsoft.org**
- **http://xmlsoft.org/XSLT**
- **http://curl.haxx.se**
- Vorbis – **http://xiph.org/vorbis/**
- **www.icecast.org/ices.php**
- **www.icecast.org/download.php**

>>

«Собственная радиостанция

Вдоволь насиделись у радиоприемника? Играйте то, что нравится вместе с *Icecast*.

ЗАЧЕМ МНЕ ЭТО?

- Живите на волне вашей любимой музыки
- Поделитесь своей музыкой и миксами с друзьями
- Размещайте собственные podcast'ы



Одна из самых замечательных возможностей при наличии широкополосного подключения к Интернету – это интернет-радио. Послушав вещание в Сети, вы вряд ли захотите вернуться к своему старому, с трудом настраиваемому на каналы аналоговому приёмнику. Сетевых станций гораздо больше, чем вы могли бы надеяться «поймать» в ограниченном спектре официальных радиочастот, причем есть они на любой вкус. Многие сетевые радиостанции довольствуются несколькими десятками слушателей – как насчёт Drone Zone («атмосферная музыка окружающего пространства, совместимая с большинством лекарственных препаратов»)?

Одна из самых популярных точек подключения к сетевым радиостанциям – **www.shoutcast.com**. Это портал для тысяч интернет-радиостанций, использующих протокол Shoutcast для передачи аудиопотока на компьютеры слушателей. Попробуйте поискать какую-нибудь песню на сайте – неважно, широко известную или нет: всегда есть шанс, что где-то в мире какой-нибудь Shoutcast-сервер исполняет именно её.

Shoutcast – проприетарный протокол, разработанный компанией Nullsoft, той самой, которая выпускает *Winamp*, один из лучших медиа-плейеров для Windows. Протокол в основном оперирует данными в формате MP3 с несколькими HTTP-командами, и чтобы прослушать такой поток, вам потребуется плейер типа *XMMS* или *Amarok*.

Потребуется также соединение со скоростью более 128 кб/с, это в два раза больше «модемной» скорости 56 кб/с, но широкополосным подключением обеспечивается легко.

«УДОСТОВЕРЬТЕСЬ, ЧТО ВЫ ИМЕЕТЕ ЮРИДИЧЕСКИЕ ПРАВА НА ВЕЩАНИЕ.»

ТРЕБОВАНИЯ:

libxml2	**http://xmlsoft.org/**
libxslt	**http://xmlsoft.org/XSLT/**
curl	**http://curl.haxx.se**
Ogg Vorbis	**http://xiph.org/vorbis/**
Ices2	**www.icecast.org/ices.php**
libshout	**www.icecast.org/download.php**
Icecast	**www.icecast.org/download.php**

Стань ди-джеем

Конечно, вы не ограничены прослушиванием чужих потоков. Вы можете запустить свой собственный Shoutcast-сервер и слушать собственную коллекцию, находясь вдали от дома, или передавать свою музыку трём-четырём слушателям в мире, одобряющим ваш музыкальный вкус. Обязательно надо оформить разрешение на воспроизведение любой имеющейся у вас музыки. Еще недавно, если вы хотели вещать на законных основаниях, нужно было подписывать соглашение с агентом каждого исполнителя. Сейчас вы можете получить единую лицензию, стоимость которой пропорциональна доходу от вашей радиостанции.

Проприетарный Shoutcast-сервер распространяется бесплатно, но есть более правильное решение – *Icecast*. Это открытый продукт, способный создавать совместимые с Shoutcast потоки, а также использовать Ogg Vorbis в качестве замены формату MP3, и для его настройки нужно выполнить лишь несколько простых шагов.

СКАЧАЙТЕ И УСТАНОВИТЕ ICECAST — ШАГ 1

Если у вас нет удобного пакета Icecast в вашем дистрибутиве (например, RPM- или Deb-файла в менеджере пакетов), вам нужно будет скачать последнюю версию исходного кода с сайта **www.icecast.org/download.php**. Скачав, распакуйте архив, перейдите в созданный каталог и запустите **./configure**, а затем **make install**, как показано в следующем примере:

```
tar xvzf ~/Desktop/icecast.tar.gz
cd icecast
./configure –prefix=/usr
make install (с правами root)
```

НАСТРОЙКА — ШАГ 2

В *Icecast* источником аудиопотока является так называемая точка доступа (mountpoint), и у каждого потока, который вы предоставляете, она должна быть своя. Для потока можно выбирать не только различную музыку, но и различную частоту дискретизации. От этой частоты и от полосы пропускания зависит качество звучания. Настройки, как и у других серверов Linux, выполняются ручным редактированием нескольких текстовых файлов. Для большинства дистрибутивов они расположены в **/usr/share/icecast**, за исключением файла **icecast.xml**, который находится в **/etc/icecast.xml**.

Откройте **icecast.xml** в вашем любимом редакторе, да хоть и в *Emacs*. Вам потребуется кое-что поменять. Убедитесь в правильности путей к другим конфигурационным файлам, особенно к файлу журнала, который обычно указывает куда-то наподобие **/var/log/icecast**. Поскольку Icecast запускается от имени обычного пользователя (не суперпользователя), нужно убедиться, что каталог **/var/log/icecast** существует и ваш пользователь *Icecast* имеет права на доступ к нему.

Например, ниже мы даём пользователю «graham» право записи в каталог лог-файлов:

```
mkdir /var/log/icecast
chown graham:users .            # в случае SUSE
chown graham:graham .           # в случае Mandriva
```

Если вы запускаете общедоступный сервер, рекомендуется создать пользователя «icecast» исключительно для целей запуска сервера. Также рекомендуется изменить пароли «source-password» и «admin-password» в файле **icecast.xml** (по умолчанию они смеха ради установлены как «hackme») и поменять параметр «hostname» на доменное имя вашей машины. Проделав всё это, протестируйте сервер, выполнив команду **icecast -c /etc/icecast.xml** от имени вашего пользователя *Icecast*. В типичном для Linux ответе, если всё работает как нужно, смотреть будет не на что.

ЧТО-НИБУДЬ ПОСЛУШАЕМ — ШАГ 3

Итак, сервер запущен. Можете проверить его работу, открыв браузер на сервере и введя адрес **http://localhost:8000/admin**. Имя пользователя будет «admin», а пароль – тот, который вы вписали в конфигурационный файл **icecast.xml**.

А теперь – волнующий момент. Введите в адресной строке вашего браузера **http://localhost:8000/test.ogg**. И... ничего не произойдет, поскольку *Icecast* ищет файл **test.ogg** в каталоге **/usr/share/icecast**. Вы можете убедиться в этом, открыв файл **error.log** (туда следует заглядывать в первую очередь, если вы наткнулись на проблему). Понятно, что нужно сделать, чтобы ваша музыка зазвучала: просто скопируйте звуковой файл из вашей коллекции в **/usr/share/icecast/test.ogg** и перезагрузите поток. Теперь вы должны услышать файл, воспроизводимый с помощью *Icecast*. Страница администрирования обновится, чтобы показать новых слушателей.



Icecast имеет собственную администраторскую веб-страницу.

МЕНЯЯ ПЕСНИ – ICES2 — ШАГ 4

Теперь всё должно работать, но проблема заключается в том, что вы можете указать только один аудио-файл. Есть более гибкое решение – сначала оно покажется вам странным: нужно запустить так называемый source-клиент для *Icecast*. Он отвечает за формирование потока данных, который *Icecast* использует как источник вещания. Есть несколько разновидностей клиентов, среди них есть и ориентированные на «живое» вещание и ди-джеев. Один из наиболее популярных – *Ices2*.

Ices2 устанавливается аналогично *Icecast*. Скачайте исходный код, запустите **./configure** и затем **make install** в распакованном каталоге. Нетрудно догадаться, что тут тоже не обойдется без загадочного конфигурационного файла. К счастью, в пакет включено несколько примеров, они находятся в каталоге **conf**. Один из них предназначен для использования в качестве источника входа звуковой карты в качестве источника (под именем **ices-alsa.xml**), но нам больше подойдёт другой, под названием **ices-playlist.xml**.

После того как вы откроете файл **ices-playlist.xml** в вашем редакторе, вам нужно будет поменять пароль, указанный между тэгами **<password>** и **</password>**. Здесь должен быть тот самый пароль администратора, который вы определили в конфигурации *Icecast*, по той простой причине, что *Ices2* будет добавлять новую точку доступа для сервера *Icecast*. Заодно нужно будет изменить имя хоста (hostname) в файле конфигурации, чтобы получать доступ к серверу по сети.

Файл конфигурации содержит настройки частоты дискретизации потока, которые следуют за тэгом **<nominal-bitrate>**. Если ваши аудио-файлы выдаются не со стандартной (44100 кГц) частотой, вам потребуется указать этот параметр.

С редактором Kate редактировать XML-файлы и Icecast, и Ices2 много проще.

РАДИО GA GA

Как вы могли заметить, в конфигурационных файлах гораздо больше информации, чем мы исследовали. Большинство радиостанций в Сети, например, используют «живое вещание», подмешивая к потоку данные со звуковой карты, прямо как настоящие ди-джеи. Это хороший способ сделать вещание менее механистичным, чем простая перетасовка play-листа. Но наш пример более полезен, если вы хотите послушать свою музыкальную коллекцию, находясь вдали от своего компьютера.

PLAY-ЛИСТ ICES2 — ШАГ 5

Ключи к предоставлению списка аудио-файлов – в файле **ices-playlist.xml**. *Ices2* использует текстовый файл, который рассматривается как простой play-лист. Это означает, что вы можете перемешивать дорожки или просто проигрывать их в определенной последовательности. Согласно конфигурационному файлу, он называется **playlist.txt**, и вам нужно будет убедиться, что файл указан правильно. В этом текстовом файле просто перечислены все композиции, которые вы хотите добавить, на отдельных строках. Например:

```
/home/graham/ogg/broadcast1.ogg
/home/graham/ogg/broadcast2.ogg
/home/graham/ogg/broadcast3.ogg
```

Наконец, вам нужно запустить *Ices2* с единственным аргументом, указывающим на изменённый конфигурационный файл:

```
ices ices-playlist.xml
```

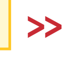
Ices2 создаст точку доступа на сервере *Icecast* и выведет аудиопоток, заполняемый из вашего play-листа. Чтобы подключиться к потоку, откройте ваш медиа-плейер и введите **http://localhost:8000/example1.ogg.m3u**. Точка доступа **example1.ogg** определяется конфигурацией *Ices2*. Теперь вы должны услышать аудиопоток через ваш сервер *Icecast*, аккуратно сформированный с помощью *Ices2*. Если ваш плейер поддерживает метаданные, вы также увидите имя исполнителя и название трека в том виде, в каком эта информация содержится в оригинальном аудиофайле.

Если вы откроете страницу администрирования сервера *Icecast*, появится точка доступа *Ices2*, вместе с метаданными трека и потока. С этой же страницы вы также сможете «сбрасывать» людей с потока и перемещать их на другие потоки, возможно, использующие более низкую пропускную способность.





Amarok **неплохо воспроизводит потоки** ***Icecast*****, но не показывает метаданные.**



>>

Запустите web-сервер

Apache – самый популярный web-сервер в мире. Давайте выясним, почему...

ЗАЧЕМ МНЕ ЭТО?

- Чтобы самостоятельно размещать сайт
- Чтобы при поддержке PHP создавать динамический контент
- Чтобы запустить RSS-каналы, wiki-страницы, форумы и другие компоненты



Управление личным веб-сервером обычно считается низшим этапом на пути становления системного администратора, но это не значит, что от него мало пользы. Истина в том, что наибольшее наслаждение от webвеб-сервера приходит не от простой установки *Apache* и создания нескольких HTML-страничек. Настоящее удовольствие – это создание динамических скриптов, включение поддержки WebDAV, «вставка» Wiki и сетевых дневников (blog), чтобы сервер заработал в полную силу.

Мы начнём с компиляции PHP и *Apache* с нуля, сделаем несколько базовых страничек, затем посмотрим, какие из существующих инструментов можно применить, чтобы добавить нашему сайту изюминку.

Первые шаги

Мы поместили на наш диск архив с *Apache 2.2*, так что скопируйте его на ваш диск и распакуйте:

```
tar xvf httpd-2.2.0.tar.bz2
cd http-2.2.0
```

Устанавливать наш сервер мы будем в **/usr/local/apache**, так что созданный при распаковке каталог вы сможете его удалить, если он вам станет не нужен. Однако желательно включить поддержку WebDAV и модуля обеспечения параллельных процессов (multi-proccess module, MPM) . Последнее очень важно, ибо ядро PHP и базовые его расширения поддерживают потоки, а, стало быть, замечательно работают в новом потоковом окружении *Apache 2.x*. Но многие унаследованные сторонние PHP-расширения небезопасны при использовании потоков, и могут вызывать серьёзные проблемы, если ваш *Apache* использует потоки при обработке запросов. Потребуйте **mpm=prefork**: тогда гарантируется, что *Apache* будет работать в режиме совместимости с версией 1.3, и использование расширений будет безопасным.

Чтобы подготовиться к такой установке, выполните следующую конфигурационную команду:

```
./configure --prefix=/usr/local/apache --enable-dav --with-mpm=prefork --enable-so
```

Затем выполните **make**, переключитесь в режим суперпользователя и выполните **make install**. Запустите *Apache* на 80-м порту (по умолчанию), выполнив команду **/usr/local/apache/bin/apachectl start**, потом запустите ваш браузер и введите **http://localhost**. Если всё работает правильно, браузер так и сообщит.

Поделиться файлами

Мы включили поддержку WebDAV во время сборки *Apache* – теперь можно делать разные замечательные штуки, например:

```
Nano conf/httpd.conf.
```

```
Найдите строку «dav» и удалите комментарий в начале строки «#include conf/extra/httpd-dav.conf»
```

Выполните следующие команды:

```
cd /usr/local/apache/htdocs
mkdir uploads
chgrp daemon uploads
cd ..
bin/htpasswd -c user.passwd admin
Откройте в редакторе файл conf/extra/httpd-dav.conf
Измените DavLockDB на «/tmp/DavLock»
Измените /usr/local/apache/uploads на /usr/local/apache/htdocs/uploads
Измените AuthType на Basic
Измените AuthName с DAV-upload на «My WebDAV Area»
Выполните bin/apachectl restart
```

Многие современные операционные системы имеют поддержку WebDAV в ядре, так что сетевые каталоги рассматриваются как локальные. Попробуем это в Gnome. Пройдите в меню Places > Connect To Server (Переход > Подключение к серверу), выберите тип сервиса WebDAV (HTTP), введите сервер «localhost», папку «uploads», имя пользователя «admin» и «My WebDAV» в качестве имени соединения, затем щелкните на Connect (Соединиться). На ваш рабочий стол будет помещена небольшая иконка; по двойному щелчку на ней у вас будет запрошен пароль,

> «НАСТОЯЩЕЕ НАСЛАЖДЕНИЕ – СОЗДАВАТЬ ХИТРЫЕ ДИНАМИЧЕСКИЕ СКРИПТЫ, ЧТОБЫ СЕРВЕР РАБОТАЛ В ПОЛНУЮ СИЛУ.»

ПОЗАБОТЬТЕСЬ О БЕЗОПАСНОСТИ С NMAP

Представьте себе: вы один из последних представителей свободной человеческой расы, в то время как другие порабощены расой злобных машин и используются в качестве источников питания. Вам нужно проникнуть в центральный компьютер, добраться до проводов и рвануться в атаку. Чем вы воспользуетесь? Конечно же, *Nmap*.

Всё верно: в фильме *«Матрица: Перезагрузка»* Тринити сканирует Матрицу на предмет доступных *OpenSSH*-серверов и взламывает их, используя реальную уязвимость. Работа *Nmap* как раз и заключается в сканировании одного или нескольких IP-адресов и выдаче информации о том, какие порты открыты и какие ОС используются. Конечно, *Nmap* нельзя рассматривать только как лучшего друга хакера, ведь это также прекрасный инструмент для анализа вашей собственной сети на предмет возможных уязвимостей.

Если вы пробуете всё, что написано в этой статье, то вы установите всевозможные программы, которые открывают порты на вашем компьютерае. Очень важно отслеживать, какие порты открыты, чтобы не запустить какой-то ненужный сервис, а также чтобы вовремя заметить, что в ваш компьютер кто-то нашел лазейку.

Мы включили последнюю версию *Nmap* в наш диск, так что распакуйте архив, перейдите в полученный каталог и выполните **./configure, make**, **su**, затем **make install**, и можно приступать.

Nmap лучше всего запускать от имени суперпользователя, потому что она использует специальные режимы соединения с TCP/IP-сокетами для обеспечения пика производительности. Итак, из-под root запустите следующую простую команду:

```
nmap -O localhost
```

По этой команде будут просканированы все порты вашей машины и выведен отчет об открытых портах. Если вы запустили *Apache*, вы увидите, что порт 80 находится в состоянии «open» (открыт), и что *Nmap* сопоставила этот порт с сервисом «http». Изучите весь список, пожалуйста: нет ли незнакомого порта? Сравните вывод с тем, что выдает ***ps aux***, чтобы увидеть, не запущена ли какая-нибудь «приблудная» программа или обычно неиспользуемый сервис. Грамотные администраторы содержат «закрытый» сервер, запретив всё, что не требуется для работы — так безопаснее.

и вы, вероятно, сможете сохранить пароль на время текущей сессии, чтобы не пришлось вводить его заново.

После ввода пароля появится новое окно, открывшееся в *Nautilus* – файловом менеджере Gnome. Теперь можно копировать все файлы, какие пожелаете, в это окно; они будут размещаться в каталоге **htdocs/uploads** вашей системы. WebDAV обладает замечательным свойством: хотя *Nautilus* рассматривает его как каталог, но если вы запустите Firefox и укажете **http://localhost/uploads**, то увидите стандартный листинг каталогов *Apache*.

Сетевые календари

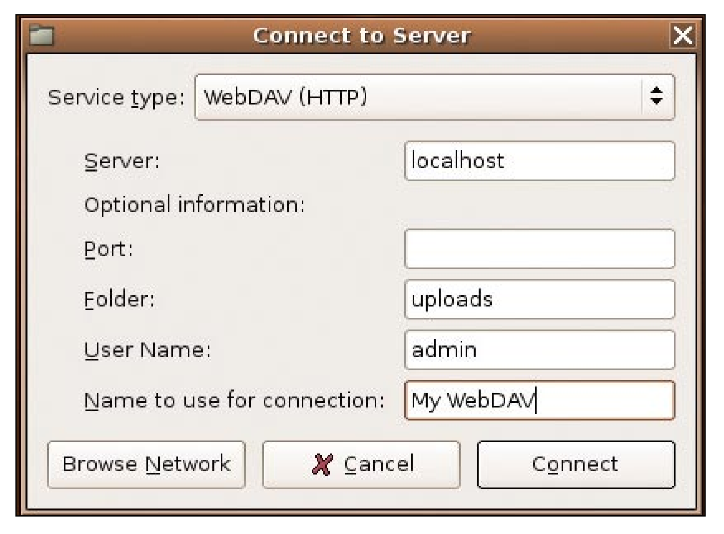
Теперь попробуем ещё одну интересную возможность: опубликуем в сети календарь с помощью приложения *Sunbird*, разработанного Mozilla Foundation. Это очень полезно, если вы работаете, скажем, над групповым проектом и хотите, чтобы каждый имел доступ к плану работ. Да мало ли причин, по которым может понадобиться календарь на сервере.

Заберите **tar.bz2**-архив с *Sunbird* с нашего диска и распакуйте его куда-нибудь, затем запустите в этом каталоге команду *sunbird*. *Sunbird* может работать и с локальными календарями, публикуя их там, где вы скажете, и с удаленными, автоматически выполняя синхронизацию.

На левой панели найдите вкладку Calendars; выберите её, и увидите локальный «домашний» календарь (Home), приготовленный для вас по умолчанию. Независимо от того, хотите ли вы использовать локальные календари или сетевые, первым делом опубликуем эту заготовку в сети.

Щелкните правой кнопкой мыши на Home и выберите Publish Entire Calendar (Публиковать весь календарь). В строке адреса, которая появится, введите **http://localhost/uploads/mycal.ics** и щелкните Publish. Введите имя пользователя «admin» и свой пароль, и ваш (пустой) календарь будет теперь доступен другим для чтения и скачивания.

Теперь заполняйте его событиями. Но это статический календарь, поэтому локально сделанные в нем изменения не отразятся в сети, пока его не опубликуют повторно, что означает лишнюю мороку. Гораздо лучше завести удалённый календарь, так что уничтожьте календарь Home (щёлкните правой кнопкой и выберите Delete Calendar), затем ещё раз нажмите правую кнопку и выберите New Calendar. Выберите Remote, затем – WebDAV. Введите **http://localhost/uploads/mycal.ics** в поле URL, затем щёлкните Next. Задайте ему имя, например, «Удаленный календарь», и цвет поинтереснее серого.



***Nautilus* воспринимает папку WebDAV как обычный каталог на вашем ПК.**

Добавим скрипты

WebDAV – очень хороший способ работать с файлами в Сети, но наш web-сервер пока что обслуживает только статические HTML-файлы.

```
tar xvfj php-5.1.1.tar.bz2
cd php-5.1.1
./configure --with-apxs2=/usr/local/apache/bin/apxs
make
su
make install
```

Данная последовательность команд добавляет модуль PHP к *Apache*, но нам ещё нужно подправить файл **httpd.conf**, чтобы сопоставить .php-файлы с интерпретатором PHP. Переключитесь в режим суперпользователя, выполнив команду **su –**, затем откройте **/usr/local/apache/conf/httpd.conf** в вашем любимом текстовом редакторе, найдите строки AddType и листайте их до тех пор, пока не увидите «AddType application/x-compress .Z». Под существующими строками AddType добавьте ещё одну:

```
AddTye application/x-httpd-php .php
```

Сохраните изменения и выйдите из редактора, затем запустите команду **/usr/local/apache/bin/apachectl restart**. Теперь создайте файл **info.php** в каталоге **/usr/local/apache/htdocs** следующего содержания:

```
<?php
        phpinfo();
?>
```

Если всё сработало правильно (код-то незатейливый), вы увидите экран, похожий на показанный справа. Теперь у вас есть всё для программирования на PHP!

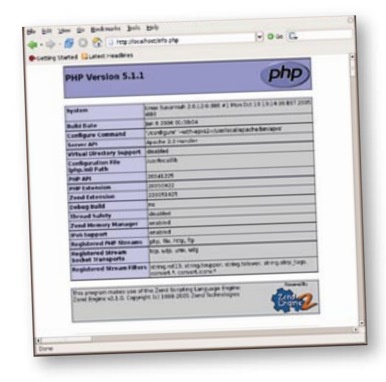
Принимаем гостей

И последнее (но не по значению!): что за радость, если на собственном сервере нельзя выделить местечко для друзей? Не бойтесь, полный доступ им давать необязательно. *Apache* включает модуль mod_userdir, который позволяет размещать веб-страницы в пользовательских домашних каталогах. Чтобы активизировать mod_userdir, потребуется только одно небольшое изменение: откройте **httpd.conf** и удалите комментарий в начале строки **«#include conf/extra/httpd-userdir.conf»**.

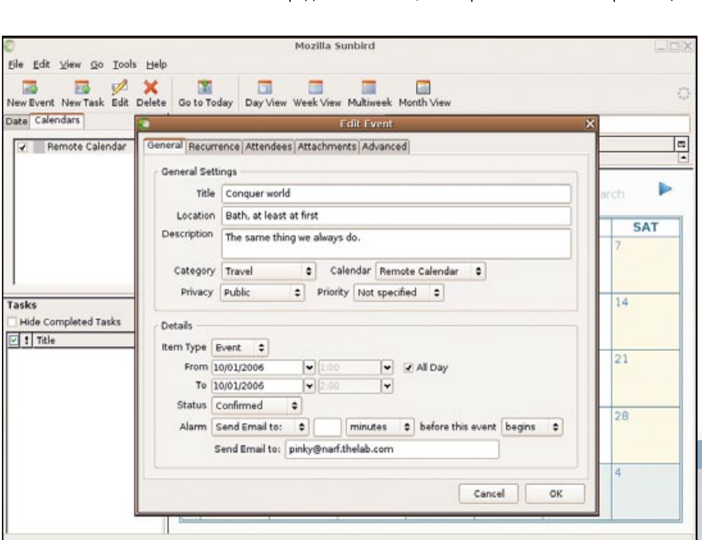
Затем перезапустите *Apache* командой **/usr/local/apache/bin/apachectl restart** и откройте терминал с правами обычного пользователя. Вы попадёте в свой домашний каталог, например, **/home/bob**. Выполните команду **mkdir public_html** – она создаст новый каталог, где *Apache* будет искать вашу домашнюю страницу. Создайте в этом каталоге файл **index.html** и поместите в него какой-нибудь комментарий. Теперь, чтобы почитать сайт, мы просто наберём в браузере **http://localhost/~username**.

Apache предоставляет пользователям все расширения, загруженные ранее, а следовательно, пользователь Bob, если пожелает, сможет использовать PHP-сценарии. Предоставление пользователю web-страницы через пользовательскую учётную запись означает, что пользователи получат полный SSH-доступ для загрузки своих файлов и управления своим пространством, а обо всём остальном позаботится *Apache*.

>>



Установка PHP удалась, но этому скрипту рано в отставку: он предоставляет много ценной информации.



Создайте событие в *Sunbird*, сохраните его в своем удалённом календаре, и *Sunbird* автоматически покажет его всем.



Строим файловое хранилище на Samba

Совместное проживание OS X, Windows и Linux – дурдом? Только не при *Samba*...

ЗАЧЕМ МНЕ ЭТО?

- Чтобы получить доступ к своим файлам с любого ПК в сети
- Чтобы предоставить ваш принтер в общее пользование
- Чтобы упростить создание резервных копий



Наиболее частая операция, выполняемая в небольших или домашних сетях – обобществление ресурсов. Общим может быть подключение к Интернету или что-нибудь попроще – например, принтер или вообще файл. Прежде чем рассылать 5 МБ-документы по всему дому через Gmail, подумайте о безопасной и простой альтернативе – вашем собственном файловом сервере.

Есть несколько способов создать его в Linux, но, вероятно, простейший и надёжнейшийсамый надежный способ – это использование *Samba*. Ко всему хорошемупрочему, Samba поддерживает протокол SMB, который используется в сетях Windows для предоставления файлов в общий доступ – так что вы сможете работать с общими файлами не только на Linux-машинах, но и на Windows, и на Mac!

Здравствуй, Samba!

Наилучший способ установить *Samba* на ваш Linux-компьютер – это воспользоваться пакетами, включенными в ваш дистрибутив: *Samba* имеется в любом дистрибутиве, претендующем на популярность. Итак, пусть ваш менеджер пакетов отыщет и установит соответствующие программы. Многие дистрибутивы разделяют клиентское и серверное ПО – убедитесь, что установлено и то, и другое. На «чистом» сервере клиент, конечно, не обязателен, но он пригодится для тестирования.

> «ДОПУСКАЙТЕ К ФАЙЛАМ ТОЛЬКО СВОИХ ИЛИ СДЕЛАЙТЕ ФАЙЛЫ НЕВИДИМКАМИ.»

После установки настает время создать конфигурационный файл. Пакет *Samba* обычно снабжен файлом настроек по умолчанию, и достаточно лишь слегка подправить его. Впрочем, вы можете начать и с нуля. Краткий пример показывает, как мало этому файлу надо (кстати, его имя его – **/etc/samba/smb.conf**):

```
[global]
workgroup = LINUX
netbios name = MYSHARE
[share1]
path = /tmp
[share2]
path = /share
comment = Some random files
```

Секция **global** обязательна и определяет имя рабочей группы (используется в Windows для объединения общих файлов-группировки ресурсов) и имя NetBIOS., пПод этим именем ваш сервер будет фигурировать на других компьютерах. По умолчанию берется имя вашего хоста (hostname).

Секции **share** в приведённом примере – это имена общих папок [частенько называемых «шарами», – прим. ред.], которые обслуживаются данной машиной. Опции не ограничиваются простой ссылкой на каталог. Вы можете допустить к общим папкам только лиц из определенного круга или даже сделать список файлов невидимым для неавторизованных пользователей. За дополнительной информацией обращайтесь к демонстрационному конфигурационному файлу или врезке «*Полезные ссылки*».

Наконец, для запуска сервера выполните команду:

```
/etc/init.d/samba start
```

Вероятно, вам захочется, чтобы этот сервис сам активизировался при каждой перезагрузке, так что либо вручную свяжите его с runlevel, либо включите в конфигурацию загрузки компьютера. Используйте для этого инструменты вашего дистрибутива.

В Linux, вы теперь можете монтировать общие каталоги и сетевые ресурсы, используя команду *smbmount*, или обычную mount с ключом **-t**:

```
mount //HOSTNAME/sharename /mnt/share -t smbfs -o user
```

В Windows и Mac вы можете использовать обычные сетевые инструменты.

Кое-что о принтере

Помните, мы вскользь отметили, что в общее пользование часто предоставляется принтер? Конфигурационный файл *Samba* и с этим способен справиться, причем несколькими способами, в зависимости от того, работает ли у вас CUPS (Common Unix Printing System) и сколько у вас принтеров. Мы будем иметь дело с простым сценарием, без CUPS (хотя CUPS, между прочим, тоже присутствует во многих дистрибутивах Linux, и это лучший способ управлять принтерами в Linux). CUPS тоже помог бы настроить сетевую печать, но раз уж запущен *Samba*, то проще поддерживать один набор файлов.

Обычно файл **smb.conf**, устанавливаемый по умолчанию, содержит следующие строки (проверьте, что они раскомментированы, т.е. символы **«#»** есть только там, где показано):

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
# чтобы разрешить «гостевой» доступ к печати
guest ok = yes
writable = no
printable = yes
create mode = 0700
#=============================
# команда печати, подробности см. выше
#=============================
print command = lpr-cups -P %p -o raw %s -r
# используются драйверы печати на стороне пользователя
```

Этот код, в общем-то, разрешает пользоваться принтерами всем. Но бывают обстоятельства, когда лучше позаботиться, чтобы на принтер покушались только штатные пользователи – в этом случае просто замените строку «guest ok = yes» на «guest ok = no».

Данные настройки всего лишь позволяют передавать данные – на клиентских машинах, естественно, должен иметься правильный драйвер принтера.

ПОЛЕЗНЫЕ ССЫЛКИ

- www.samba.org/samba/docs/man/Samba-HOWTO-Collection
- http://hr.uoregon.edu/decidrl/samba.html
- http://samba.anu.edu.au/cifs/docs/what-is-smb.html
- ftp://ftp.samba.org/pub/samba/docs/Samba24Hc13.pdf

Запрягаем базу данных

Ваш собственный сервер баз данных *MySQL* в подкрепление к *OpenOffice.org Base*.

Баз данных в мире заметно больше, чем web-серверов. Они гораздо дольше окружают нас, содержат на порядок больше данных и поставляют больше критичных сервисов, чем все прочие. А почему бы вам не завести собственную базу данных? В отличие от серверов *Samba* и *Apache*, серверы баз данных не гонятся за славой – запустите такой, и он будет сидеть себе потихоньку в фоновом режиме, дожидаясь запроса.

Web-хостьеры обычно предоставляют базы данных весьма скромных размеров (заполучить 100 МБ считается удачей) – вот вам и повод сделать сервер самому. Потратив некоторое время на освоение SQL, вы сможете настроить ваш сервер и запросто предоставлять его также и вашим друзьям.

Ваше руководство по установке

Мы продемонстрируем вам, как настроить сервер *MySQL*. *MySQL 5.0* имеется на нашем диске, так что скопируйте его, распакуйте и перейдите в созданный каталог, где выполните **./configure --prefix=/usr/local/mysql**, затем **make**, переключитесь в режим **root**, и наконец, запустите **make install**. Теперь перейдите в каталог **/usr/local/mysql**, выполните там команду **bin/mysql_install_db**, затем **chown -R mysql var**. Итак, *MySQL* установлена. Выполните команду **bin/mysqld_safe &**, чтобы запустить ваш *MySQL*-сервер.

По умолчанию *MySQL* не устанавливает пароль для своего суперпользователя, так что вам нужно ввести несколько команд от имени непривилегированного пользователя:

```
# Входим на сервер
/usr/loca/mysql/bin/mysql -u root mysql
# Зададим пароль для root - желательно, чтоб угадать его было нелегко!
UPDATE user SET Pasword = PASSWORD('frosties') WHERE User = 'root';
# Создайте базу данных, которая нам понадобится в дальнейшем
CREATE DATABASE lxfdb;
# Добавьте учетную запись непривилегированного пользователя
INSERT INTO user (Host, User, Password) VALUES ('localhost', 'lxf', PASSWORD('orangutan'));
# Предоставьте новому пользователю доступ к созданной базе данных
GRANT ALL PRIVILEGES ON lxfdb.* TO lxf;
# Обновите привилегии MySQL
FLUSH PRIVILEGES;
# Выйдите из MySQL; негоже лазить везде с правами root
exit;
```

Мы вернулись к нашей командной строке, теперь осталось убедиться, что новая учётная запись работает. Введите:

```
# Входим под именем нашего пользователя
/usr/local/bysql/bin/mysql -u lxf -p lxfdb
# Посмотрим список имеющихся таблиц (пока у нас ничего нет)
SHOW TABLES;
exit;
```

Если всё в порядке, значит, учётная запись создалась нормально.

Чтобы создать учётные записи ещё для кого-нибудь, просто повторите команду **INSERT INTO**, замените имя пользователя и пароль новыми значениями, плюс установите значение Host в «%» (предоставление доступа к локальному хосту извнес удаленных хостов). Вы можете создать сколько угодно баз данных и предоставлять пользователям различные уровни доступа. *MySQL* работает на порту 3306, так что вам нужно будет открыть доступ к нему на вашем брандмауэре, чтобы другие могли подключаться беспрепятственно.

Жизнь-служение

По общему признанию, серверы баз данных не богаты приложениями, преимуществами которых можно воспользоваться прямо с ходу, но некоторые всё же существуют. Например, можно настроить *OpenOffice.org Base* на взаимодействие с вашим новым сервером для всех её баз данных – вы получите ту же самую дружественную оболочку, но надёжно подкрепленную прекрасно масштабируемым *MySQL*.

Если душа просит чего-то более «продвинутого», то как насчет возможности доступа к вашим данным из любой точки? Лучше всего вооружиться программой типа *PhpMyAdmin*: это основанный на PHP инструмент администрирования *MySQL*-сервера, позволяющий вам создавать данные и управлять ими через Интернет. Будьте осторожны с правами доступа; лучше всего установить отдельную копию *phpMyAdmin* для каждого пользователя, чтобы быть уверенным в надлежащем разграничении прав.

Многие скрипты используют *MySQL* для хранения своих данных (wiki и дневники, пожалуй, наиболее яркие представители), но, бесспорно, лучшее применение *MySQL* – это разработка собственных PHP-скриптов на базе *MySQL*.

>>

МЕЖСЕТЕВЫЕ ЭКРАНЫ

Запуская собственный сервер, вы вынуждены «продырявить» свой брандмауэр (firewall), чтобы ваш интернет-сервис мог работать, потому что брандмауэр зачастую блокирует все входящие соединения – беспрепятственно проходят только исходящие.

Чтобы открыть порт на вашем брандмауэре для какого-либо сервера, например, *Apache* (HTTP), вам нужно сначала узнать, какой порт и протокол этот сервер использует. Эту информацию обычно можно найти в документации сервера; пригодится также наша небольшая таблица.

И SUSE, и Mandriva используют собственные инструменты настройки брандмауэра, доступные из панели управления. В SUSE, например, откройте *Yast*, выберите «*Security and Users*» и щелкните на кнопку «Firewall». Появится новое окно, и для открытия порта просто щелкните на вкладку «Allowed Services» слева.

SUSE поставляется с несколькими предварительно настроенными сервисами, которые можно просто выбрать из выпадающего меню «Services to Allow», либо добавить ваш собственный, щёлкнув по кнопке «Advanced». Если вы предпочитаете действовать самостоятельно, то *Guarddog* (**http://www.simonzone.com/software/guarddog**) – прекрасная графическая оболочка для настройки вашего брандмауэра.

СЕРВИС	ПРОТОКОЛ	ПОРТ
SSH	TCP	22
HTTP	TCP	80
Squid	UDP, TCP	3128, 3130
Jabber	UDP, TCP	5222, 5269
VNC	TCP	5800+, 5900+
BitTorrent	TCP	6881-6889, 6969

ЗАЧЕМ МНЕ ЭТО?

- ✓ Чтобы сэкономить деньги на хостинге баз данных
- ✓ Чтобы получить прочную основу для графических систем баз данных
- ✓ Чтобы наконец изучить SQL



Создаете новую базу данных в *Base*? Это диалог для подключения к вашему новому *MySQL*-серверу. Учтите, у вас должна быть установлена Java.

Общайтесь при помощи Jabber

Соберите друзей поболтать через ваш личный сервер обмена мгновенными сообщениями.

ЗАЧЕМ МНЕ ЭТО?

- ✓ Чтобы держать свои сообщения в секрете
- ✓ Чтобы сохранять все важные «диалоги»
- ✓ Чтобы запустить собственную страничку сообщества
- ✓ Чтобы избавиться от рекламы



Если не считать голосовое общение, то мгновенные сообщения (IM) лучше всего подходят для проведения групповых дискуссий. Они заслуженно популярны, и существует ряд конкурирующих протоколов, серверов и клиентов для IM. Поскольку мы используем Linux, мы можем выбрать лучшее – по нашему мнению, это *Jabber*.

Jabber – не название какой-то программы как таковой. Это обозначение протокола, поддерживающего взаимодействие клиентов и серверов для создания чата. Вообще-то протокол называется не Jabber, более точное его наименование – XMPP, он поддерживает открытый стандарт для сообщений. Однако довольно путаницы, перейдем к делу.

Протоколами Jabber пользуются несколько серверов. Некоторые из них коммерческие, хватает и открытых; одни написаны на Java, другие – на C, и т.д. Со списком различных вариантов можно ознакомиться на сайте JFSF **www.jabber.org/software/servers.shtml**. Мы выбрали Wildfire, сервер от Jive Software, по трём причинам – этот сервер:

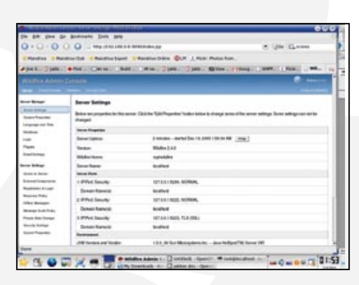
- свободное ПО, распространяемое под лицензией GPL;
- основан на Java и является кросс-платформным (Mac/Windows/Unix);
- способен работать как автономно, так и в составе сервера приложений.

Есть и четвёртая причина: его очень просто настраивать и использовать! Чтобы раздобыть последнюю версию исходных кодов и RPM-пакеты, зайдите на **www.jivesoftware.org/wildfire/** или возьмите версию с нашего диска, которую будем использовать мы. Установка RPM-пакета совершенно стандартная, так что у вас все получится (сервер будет поставлен в **/opt/wildfire**). Tar-архив содержит статически собранную версию, которую вы можете распаковать и скопировать в **/opt** или другое место. Запустите сервер:

```
/opt/wildfire/bin/wildfire
```

Дальнейшую настройку может выполнить сам сервер. Наберите в адресной строке адреса вашего браузера **localhost:9090** (или укажите имя сервера, если вы выполняете настройку удалённо). Углубившись в опции (для простоты выберите встроенную базу данных – Embedded Database), вы за несколько шагов достигнете главной панели администрирования. Здесь доступна уйма опций, но в первую очередь надо бы завести нескольких пользователей и протестировать систему. Ваш сервер – частный и закрытый, поэтому вы, вероятно, исключите регистрацию извне и будете создавать нужных пользователей самостоятельно.

Чтобы сервис *Wildfire* запускался при старте системы, в каталоге **/bin/extra** предсмотрен скрипт *wildfired*. Для его запуска на вашей системе, отредактируйте строку, начинающуюся с **export WILDFILRE_HOME=**, чтобы она указывала на каталог установки, затем скопируйте его в **/etc/init.d** и сделайте ссылку из каталога соответствующего уровня исполнения (runlevel). Точный способ сделать это зависит от вашего дистрибутива.



Web-инструмент администрирования от Wildfire упрощает управление.

Вот и всё. Пользователи смогут самостоятельно подключаться к вашему серверу и беседовать, пока фирма не обанкротится. Кстати говоря, в корпоративной сети вы можете подсоединить *Wildfire* к LDAP – учётные записи для всего персонала будут получены автоматически! (См. **www.jivesoftware.org/builds/wildfire/docs/latest/documentation/ldap-guide.html**).

Выбирайте клиент...

Существует множество клиентов, способных работать с Jabber-серверами – поскольку это открытый стандарт, их достаточно легко подключить. Вот несколько популярных совместимых клиентов для основных операционных систем:

- Linux: *Gaim, Kopete, Gush, Gabber*.
- Mac OS X: *iCat, Gush, Gabber*
- Windows: *Miranda IM, Trillian Pro, IRJabber, Gush*

Не забывайте, что вам нужно создавать учётные записи на сервере или разрешить клиентам регистрироваться самостоятельно. Приятного общения!

НАСТРОЙКИ SYSCTL

Ядра многих дистрибутивов настроены на баланс между серверным и «настольным» применением. Но если вы запускаете сильно загруженный сервер, то можно подстроить некоторые опции ядра для получения прироста скорости при помощи утилиты *sysctl*, которая изменяет параметры ядра «на лету». (Вы можете также сохранить настройки в **/etc/sysctl.conf**). Введите **sysctl -a**, чтобы увидеть все параметры.

Ряд опций могут увеличить производительность сервера, их следует вводить от имени root:

```
sysctl -w kernel.threads-max=65526
sysctl -w fs.file-max=8192
```

Эти команды устанавливают число активных потоков, которое ядро может обслуживать, и число файлов, которые можно открыть одновременно. Увеличьте эти значения, если в лог-файлах появятся сообщения о нехватке данных ресурсов.

Следующая опция увеличивает объем памяти, выделяемой для сетевых буферов отправки – если нужно обслуживать большие файлы, то это значительно ускорит машинуработу (web, FTP, *Samba* и т.д.):

```
sysctl -w net.core.wmem_
max=1048576
```

Далее, установите число соединений, которые сохраняются в ожидании подтверждения клиентом. Обычно это число – 1024; увеличьте его, если ваш сервер часто оказывается перегружен соединениями:

```
sysctl -w net.ipv4.tcp_max_syn_backlog=4096
```

Последняя опция определяет, как страницы памяти ядра сохраняются на диске (в виртуальной памяти), когда объем ОЗУ недостаточен. Обычно этот параметр устанавливается в значение 3. Более высокое значение увеличит время отклика настольных приложений (нехорошо!), но может быть полезно на сервере, где общая пропускная способность важнее, чем мгновенный отклик:

```
sysctl -w vm/page-cluster=8
```

ПОЛЕЗНЫЕ ССЫЛКИ

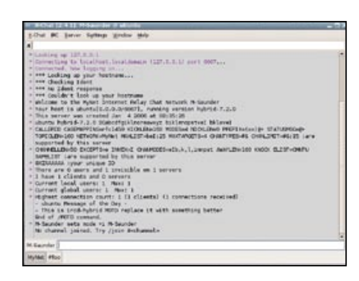
- www.habber.org
- www.jivesoftware.org
- http://psi-im.org/psi.affinix.com
- www.xmpp.org
- http://en.wikipedia.org/wiki/Jabber

Запустите собственную IRC-сеть

Откройте чат-салон на IRC-сервере – можно обсуждать открытое ПО, проекты Linux, судейство в Турине…

Internet Relay Chat (IRC) – предшественник всемирной паутины. В далёком 1988-м году программист Яарко Ойкаринен (Jarkko Oikarinen) создал альтернативу общению через доски объявлений, разработав IRC – простой и быстрый протокол, основанный на обычном тексте, который позволил пользователям во всём мире подключаться к сети серверов и беседовать в режиме реального времени. Благодаря простоте протокола IRC, даже программисты-любители могли разработать собственные клиенты – так что довольно быстро система уже насчитывала тысячи пользователей.

Сегодня рынок захвачен преимущественно чат-салонами, основанными на web (типа Yahoo Chat), но IRC всё ещё очень популярен среди опытных пользователей. Основные IRC-сети, EFnet и Freenode, размещают тысячи «каналов» (аналог чат-салона) по темам от разработок на Perl до воскрешения мёртвых. IRC выработал свои собственные обычаи и нормы этикета, и пользователи, которые хамят и высказывают неуважение к правилам, рискуют нарваться на довольно жёсткие санкции. Это диковатый и эксцентричный мир – но только не в вопросах функционирования, и один из лучших способов поддерживать связь с людьми.



Первое подключение к личному IRC-серверу через *X-Chat*. Сначала настройте каналы, а потом созывайте собеседников.

Зачем запускать IRCD?

Демон IRC, известный также как IRC-сервер, позволяет пользователям подключаться к вашей машине через своих клиентов и взаимодействовать с вами как с хостом. Вы можете запустить автономный IRC-сервер для внутреннего пользования или присоединиться к большой сети, чтобы распределить загрузку – если у вас, к примеру, есть право на присоединение вашего сервера к Freenode, то вы сможете обслуживать пользователей, которые будут общаться с пользователями других Freenode-серверов. Так… а свой-то зачем настраивать? По нескольким причинам: вы можете работать над программным проектом и пожелать работать с собственными каналами, чтоб самому устанавливать правила (вместо создания канала в чужой сети). Или, возможно, вы хотите общаться с друзьями «частным» образом, подальше от различных «асек» и основанных на web чат-салонов. А может быть, вы просто хотите поддержать существующую IRC-сеть, добавив к ней ещё одну лошадиную (в смысле, серверную) силу.

Многие нынешние IRC-серверы являются переработкой исходного кода IRCD, с набором функций, добавленных поверх него. Некоторые были созданы с нуля. Ниже мы приводим три шага, показывающие, как собрать, настроить и запустить *IRCD-Hybrid*, один из наиболее гибких серверов IRC, широко использующийся в сети EFnet.

Когда ваш сервер будет запущен, вашим пользователям понадобятся IRC-клиенты, чтобы подключиться к нему. Пользователям KDE следует попробовать прекрасный *Konversation*, в то время как фанаты Gnome/GTK могут выбрать не менее удачный *X-Chat*. Для работы в командной строке хорошим выбором будут *Irssi* и *BitchX*, и есть даже web-оболочка в виде *CGI:IRC*. Полный список можно найти на **www.freshmeat.net**.

ЗАЧЕМ МНЕ ЭТО?

- Чтобы общаться с гарантией «приватности»
- Чтобы получить полный контроль над тем, кто и для чего использует ваш сервер
- Чтобы связываться со своими друзьями в существующих сетях



ЗАВЕСТИ IRCD-HYBRID… ШАГ 1

Вы можете найти его на нашем диске в разделе **Server** или на сайте **www.ircd-hybrid.com**. Распакуйте **ircd-hybrid-7.2.0.tgz**, перейдите в созданный каталог и введите **./configure**. Здесь нет скрытых зависимостей, и опции, выставленные по умолчанию, превосходны. Но если вы пожелаете изменить максимальную длину имени пользователя или наименования канала, это можно сделать, используя ключи **--with-nicklen** и **--with-topiclen** соответственно в команде **./configure**. Введите **make** и **make install** (как root) для установки в **/usr/local/ircd** по умолчанию.

НАСТРОИТЬ… ШАГ 2

Перейдите в **/usr/local/ircd/etc/** и скопируйте файл **example.conf** под именем **ircd.conf**. Теперь откройте ircd.conf в вашем любимом редакторе, и познакомьтесь поближе с его форматом. Задайте имя вашего сервера и его SID (уникальный идентификатор, например, «8XZ») в секции «serverinfo» в самом верху. Вам нужно также найти и удалить строку **havent_read_conf**, которая предотвращает запуск сервера с настройками по умолчанию – это будет на строке 1073. Для начала вполне достаточно; позже вы сможете почитать описания других опций.

ПУСК! ШАГ 3

Введите команду **/usr/local/ircd/bin/ircd -logfile ~/ircd.log**, она запустит сервер (и будет записывать информацию о любых проблемах в файл **ircd.log** вашего домашнего каталога). Теперь вы можете подключиться к вашей машине локально (по адресу 127.0.0.1) с помощью IRC-клиента, создать каналы и установить права доступа. Затем сообщайте друзьям ваш IP-адрес или доменное имя, и они смогут подключаться через Интернет (убедитесь, что порт 6667 не заблокирован брандмауэром). Если вы хотите присоединиться к другой IRC-сети, свяжитесь с её администраторами – как правило, вы можете найти их адреса электронной почты на сайте сети или проекта. Наконец, более полную информацию по использованию IRC можно получить на **www.irchelp.org**.

НУЖНА ПОДДЕРЖКА?

Наша прогулка по IRC была последней в представленном материале. Мы надеемся, что вы попробуете хотя бы одну из предложенных идей. Прыжок на первую ступеньку администрирования сервера совершить непросто, так что если вы сражаетесь с пошаговыми инструкциями или нуждаетесь в совете, что дальше, мы рекомендуем вам посетить www.linuxforum.ru. Даже если вы просто хотите пожаловаться, что ваш любимый сервер обделили вниманием, мы все равно ждем вашего отклика! LXF

Учебники >>

Наши эксперты помогут вам с любым приложением Linux

ВИРУСЫ МОЗГА VS CLAMAV



Евгений Балдин Хорошо рисует, потому что знает *MetaPost*.

В своей жизни мне довелось прочитать довольно много страниц самого разнообразного текста. В большинстве своём, прочитанные символы не оставили о себе никаких воспоминаний, но временами попадается нечто, что заставляет задуматься. Статья Ричарда Докинза «Вирусы мозга» подкинуло моему разуму пищи для размышления. Перевод этого произведения на русский язык можно найти в сети.

Мозг человека с точки зрения «идеи» является средой размножения, как и биологическая клетка с точки зрения вируса или компьютер с точки зрения программы. Идеи, как и вирусы, могут быть полезными (предустановленными производителем или школой) и вредными (подхваченными на улице). Идеи могут бороться, а могут сотрудничать, могут помогать, а могут паразитировать.

Головы членов сообщества Open Source являются замечательным полем деятельности для идей. Непримиримые, можно сказать, «религиозные» течения (vim/emacs, KDE/Gnome, xine/mplayer и т.д.) не просто сосуществуют в одном дистрибутиве - они являются источниками кода друг для друга.

Здесь идеи не умирают. Это прямое следствие свободы.

info@linuxformat.ru

КОД В ЭТОМ РАЗДЕЛЕ

Печатать в журнале код не так легко, но мы надеемся, что некоторые простые правила позволят избежать недоразумений. Если строка кода оказывается слишком длинной, она переносится на следующие, причем все они обрамляются синим прямоугольником:

```
procedure
TfrmTextEditor.mniWordWrapClick
(Sender: TObject);
```

В противном случае, между строками будет небольшой промежуток:

```
  mniWordWrap.Checked := false
end;
```

Как правило, весь код можно найти на прилагаемом к журналу диске.

В ЭТОМ МЕСЯЦЕ ИЗУЧАЕМ...



СОВЕТ МЕСЯЦА

КРЭШ-ТЕСТ

Случается, фоновый процесс, на который вы рассчитываете, тихо умирает. Это чревато неприятностями, а обнаружить проблему бывает непросто. Например, MythTV использует фоновый процесс для записи телепрограмм. Если демон «упадет», вы ничего не получите. Конечно, в идеале программа не должна «вылетать», но если это все же случается, есть способ настроить ее автоматическое «оживление».

Суть в том, чтобы запускать процесс не напрямую, а из простого скрипта. Если программа завершается, скрипт вызывает ее вновь. Для этого служит простой цикл:

```
while true;
do
    mythbackend;
done
```

Если *mythbackend* возвратит управление, то будет запущен на следующей же итерации.

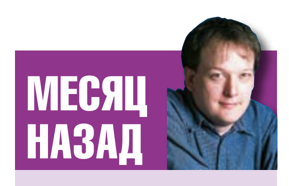
Сценарий можно улучшить, добавив проверку того, запущен ли *mythbackend* перед стартом первой копии. Для этого нужно просто узнать число процессов mybackend, профильтровав вывод команды **ps** утилитой *grep* и посчитав число строк при помощи **wc**. Если *mythbackend* уже запущен, число процессов будет больше нуля, что является условием для оператора **if**:

```
#!/bin/bash
mcount = ‘ps ax | grep “mythbackend” |
grep -v grep | wc -l’
if [$mcount == 0]; then
   #! Insert ‘while true’ code here
else
   exit 1
fi
```

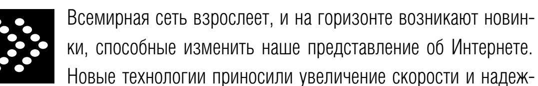
ПЕРВЫЕ ШАГИ СЕРИЯ «LINUX ДЛЯ НОВИЧКОВ»

Web 2.0 Новые инструменты Паутины

Энди Ченнел (Andy Channelle) поможет взять все от блогов, фото-сайтов и прочего.



Мы преодолели «страх сцены» благодаря Impress – мастеру презентаций в составе OpenOffice.org.



Всемирная сеть взрослеет, и на горизонте возникают новинки, способные изменить наше представление об Интернете. Новые технологии приносили увеличение скорости и надежности соединений и новые средства поиска, но пользовательская часть интерфейса оставалась на прежнем уровне: присмотритесь к вашему браузеру – и увидите, что он недалеко ушел от *Mosaic*, первого графического интернет-обозревателя, выпущенного в марте 1993 года Национальным центром суперкомпьютерных приложений (NCSA). Несмотря на всяческие улучшения, web-браузер по-прежнему в основном отображает HTML-страницы и запоминает историю посещений сайтов. Однако ситуация меняется. Разработчики экспериментируют с новыми возможностями, чтобы исполнить свое давнее обещание – сделать Интернет по-настоящему интерактивной средой.

Данное руководство посвящено интернет-приложениям нового поколения (так называемому Web 2.0), которые работают не на вашем компьютере, а удаленно. Это могут быть программы просмотра изображений, создания закладок, или просто публикации собственных мыслей на весь мир (или только для себя).

Конечно, каждый, кто пользуется Yahoo Mail, Hotmail или другими почтовыми службами, знаком с подобными приложениями уже много лет, но последние инновации в сетевых службах намного расширили их возможности. Лучшим подтверждением перспективности новинок служит тот факт, что гранды сетевого бизнеса сейчас активно скупают фирмы Web 2.0, как будто и не было кризиса дот-комов… Назло им, рассмотрим локальные программы для взаимодействия с новоявленными чудесами.

ЧАСТЬ 1 – СЕТЕВЫЕ БЛОГИ

Персональные web-страницы существуют давным-давно, но лишь с появлением служб типа Blogger и LiveJournal владельцы ПК получили простой способ довести свои идеи до всего мира – по крайней мере, компьютерного. Настройка учетной записи в этих службах (или им подобных, доступных через быстрый поиск) невероятно проста.

Однако нас интересуют инструменты, позволяющие добавлять записи в блоги (сетевые дневники) без обязательной регистрации на сайте. Самое простое решение – расширение *Blog this* для *Firefox*, работающее с сервисом Google Blogger. Расширение можно установить, выбрав в *Firefox* Инструменты > Расширения > Загрузить расширения, или скачав с **https://addons.mozilla.org/extensions/** в разделе Blogging. Может понадобиться добавить адрес этого сайта в список разрешенных URL для того, чтобы *Firefox* мог устанавливать с него расширения (если вы еще ничего не загружали для *Firefox*). Сделать это можно с помощью желтой панели, появляющейся в верхней части окна браузера, когда вы пытаетесь что-либо установить из интернета.

Как следует из названия, *Blog This* идеально подходит, если вы хотите оставлять свои заметки на сайтах. Для запуска расширения, зайдите на страницу, нарывающуюся на комментарий, и дважды щелкните мышью по любому объекту, затем выберите соответствующий апплет. С его помощью вы можете авторизоваться, после чего появится новое

окно с панелью инструментов и большим текстовым полем для ваших излияний. Панель инструментов содержит приличный набор средств форматирования текста и выпадающее меню с перечнем активных учетных записей; если вы одновременно поддерживаете связь с несколькими блогами под одним именем, здесь вы сможете выбрать, где поместить ваше сообщение. Написав послание, нажмите кнопку Publish, и ваш текст вместе со ссылкой на оригинальный материал сайта будет отослан на блог. Предусмотрена возможность сохранить комментарий как черновик и отредактировать его позднее. Для отмены комментария просто закройте его окно.

Копаем глубже

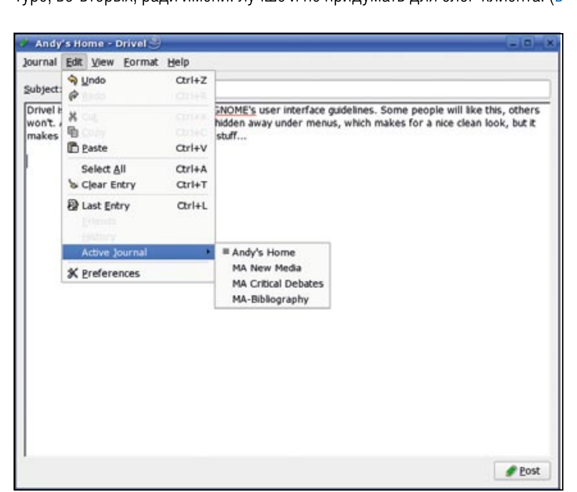
На LiveJournal нет эквивалента *Blog This*. Зато на основе *Firefox/Mozilla* существует отдельное приложение, которое сделает всю работу – *Deepest Sender* (**http://deepestsender.mozdev.org/**). Как и раньше, его можно установить в последнюю версию *Firefox*, но на этот раз оно будет доступно из меню Инструменты. Замечательная особенность программы – авторизовавшись в ней, вы сможете пользоваться полноценным клиентом LiveJournal. Пользователи могут обновлять свою музыку, выражать эмоции, размещать картинки, устанавливать логотип пользователя или сообщества и квалифицировать сообщения как публичные/приватные. Именно такого Интернет-приложения пользователи *Firefox* (а до этого – *Mozilla* и *Netscape*) ждали много лет. Как и в *Blog This*, здесь существует возможность выбрать, в какой именно журнал LiveJournal следует поместить вашу запись.

Для полной интеграции с браузером, встройте *Deepest Sender* в качестве боковой панели *Firefox* или *Mozilla* – и у вас всегда будет мгновенный доступ к журналам. Последняя версия программы может похвастаться значительным достижением – доступом к *Blogger* и *WordPress* (еще один блог-сервис). Чтобы добавить программу в левую часть окна браузера, выберить пункт меню Вид > Боковая панель > Deepest Sender. Если вы хотите ограничиться только одним инструментом ведения журнала, выбирайте этот.

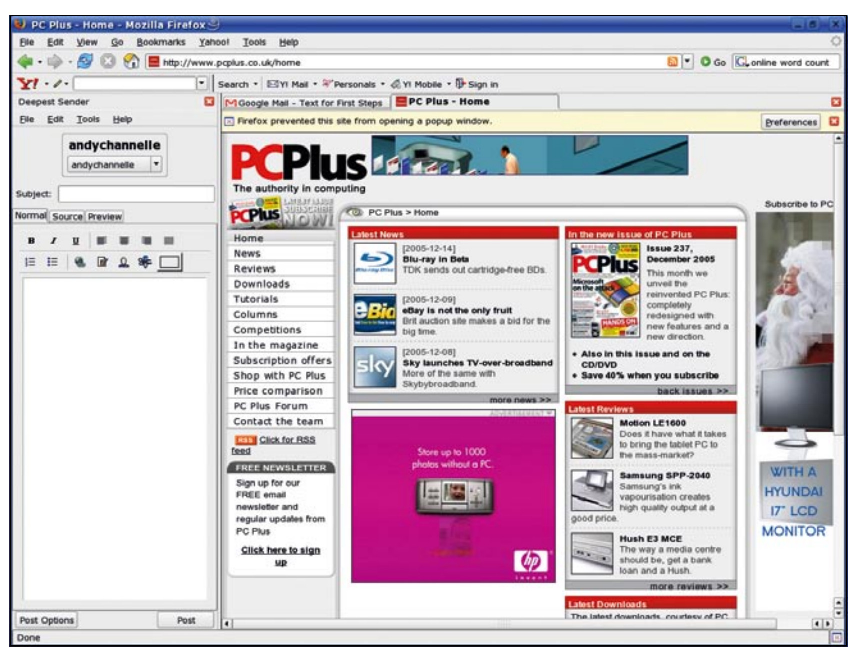
Прямые рассылки

Очевидно, что для ведения сетевого журнала нужно сначала запустить web-браузер. Однако многие владельцы блог-служб располагают своими API – интерфейсами разработчика, с их помощью почти каждый может создать собственное приложение, способное размещать сообщения в Blogger и LiveJournal. Странно, что для KDE нет «родного» приложения или апплета для работы с первым, тогда как Kluje (и кто только придумывает эти имена?) неплохо сцепляется со вторым.

В противоположность KDE, рабочий стол GNOME располагает как минимум тремя программами (включая апплет для панели задач) прекрасно работающими с большинством популярных блог-сервисов. Я выбрал *Drivel* по двум причинам: во-первых, он включен во многие дистрибутивы и поддерживает Blogger, LiveJournal, Advogato и Movable Type; во-вторых, ради имени: лучше и не придумать для блог-клиента! (в переводе с английского, «drivel» – чушь, пустая болтовня, – прим.ред.) Правда, каждая опция упрятана разработчиком в отдельное меню, и к интерфейсу программы придется привыкать.



Классное имя, классная вещь. *Drivel* всем хорош.



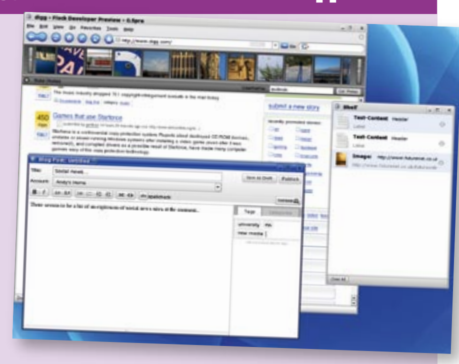
Блог всегда, блог везде – ну как не завести *Deepest Sender* на боковой панели?

После запуска *Drivel* вы увидите разочаровывающе-пустое окно: строка меню, поле для указания темы и огромное пугающее пространство, ожидающее ввода вашего текста. Можете сразу ввести тему и набирать сообщение, но если вы возжелали стилевого оформления текста или просто хотите поместить текст в другой журнал, то придется покопаться в программе. Настройки шрифта, вставка ссылок и изображения находятся в меню Format, в то время как опция Acrive Journal (Текущий журнал), позволяющая выбрать, в какой из многочисленных блогов попадет ваше сообщение, расположена (по непонятной причине) в пункте Edit (Правка). В меню имеются разнообразные опции, включая проверку правописания и блог-монитор Technorati, который сообщит вам о каждом обновлении вашей записи. Интерфейс этой программы действительно стоит изучить.

>>

FLOCK: БРАУЗЕР НОВОГО ПОКОЛЕНИЯ

НА ДИСКЕ



Интернету будущего нужен web-браузер будущего, и именно над ним работают авторы проекта *Flock*. Прихватив нутро от *Firefox*, разработчики «привинтили» к нему новую тему и дополнительные функции для пользования web-службами типа Delicious, Flickr и Blogger. Например, закладки не обязательно хранить локально на компьютере – их можно отправить на учетную запись Delicious, снабдив при необходимости тэгами и комментариями.

Новая панель в верху окна может по вашему желанию отображать содержимое вашего или чужого фотожурнала *Flickr* или список активных блогов, достигаемых одним щелчком мыши. Панель Blogger имеет функцию «Поместить прямо в журнал», так что вы можете перетащить мышью ссылку, страницу или изображение на панель, после чего откроется редактор блога (весьма похожий на Blog This для *Firefox*) с отформатированными ссылками и готовый к старту.

Очень полезна утилита Shelf (полка), имеющая собственное окно. Ссылки, изображения и цитаты из текстов можно перетаскивать в это окно для дальнейшего употребления. Особенно приятно, что все хранящиеся на «полке» материалы никуда не пропадают и исчезают только после удаления явным образом – это особенно удобно для сетевых исследований, шопинга и прочего. Наконец, во *Flock* есть отличные RSS-инструменты, например, «живые» закладки и функция агрегации «на лету».

Не обошлось и без «но»: программа пока на раннем этапе разработки и иногда ведет себя немного «чудаковато». Осторожных просим откинуться на спинку кресла и подождать, пока функции *Flock* не появятся в основном релизе *Firefox* или чего-нибудь еще.

<<

ЧАСТЬ 2 – ИЗУЧАЕМ FLICKR ПОДРОБНЕЕ

Картинка лучше описания, хоть бы и в тысячу слов, и если вы мыслите образами, то Flickr станет для вас хорошей альтернативой перенасыщенным текстом блогам. Как и на сайтах блогов, работа с Flickr начинается с настройки учетной записи. К счастью, фирма (ныне в собственности Yahoo) предлагает бесплатный хостинг на Flickr в размере 20 МБ в месяц. Большинство людей это устраивает, а если вы поддадитесь страсти фотографировать дорожные знаки, паучков и все на своем пути, подумайте о платной учетной записи «Pro», которая стоит 24.95$ в год.

Flickr выделяет пространство для фотографий каждому, причем можно устанавливать различные уровни доступа – например, разрешить смотреть семейные фото только членам вашей семьи. Программа также снабжает изображения «бирками», это удобно при поиске. Можете поискать, допустим, всевозможные виды Лондона, или всех пауков, или все фото Проекта Рая. Как я часто говорю в данном цикле статей, правильный результат можно получить, только заранее грамотно прокомментировав ваши фотографии.

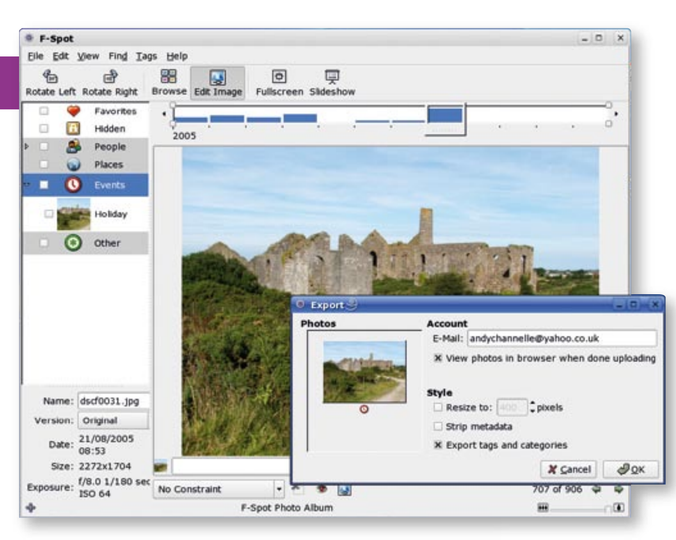
Прочие функции Flickr включают слайд-шоу, коллективную фотосессию (можно посмотреть, как одно и то же событие, например, свадьба, сфотографировано разными людьми) и Organizr, позволяющий менять порядок снимков, вращать их и создавать альбомы, но пока не умеющий редактировать сами фотографии. Также, как с блогами (см. выше), в этом руководстве нас интересуют способы закачки фотографий на сайт, а не манипуляция изображениями, которые уже в сети.

Рассылка изображений по E-mail

Самый платформо-независимый способ закачки фотографий в интернет – пересылка файлов по электронной почте с помощью обычного почтового приложения.

Когда вы настроите учетную запись Flickr, у вас появится почтовый ящик типа **вашеимя@photos.flickr.com**. Отправленная по этому адресу фотография как по волшебству попадет в вашу коллекцию. Не забывайте, что новое изображение наследует стандартные настройки вашей учетной записи, поэтому мы кое-что исправим.

Сперва настроим права доступа. Кроме режима «Личные», когда фотографии можете увидеть только вы, имеются варианты «Семья», «Друзья» и «Публичные». Чтобы изменить права доступа для фотографий, необходимо отредактировать первую часть e-mail-адреса. Так, посланные по адресу **вашеимя+friends@photos.flickr.com** фотографии будут доступны для всех, кто входит в ваш список друзей на Flickr. Аналогично, адрес **вашеимя+family@photos.flickr.com** позволит смотреть фотографии членам вашей семьи. Для комбинации доступа друзья+семья воспользуйтесь параметром **ff**, а для личного доступа – **private**.



F-Spot **– крутой фотоменеджер на основе Mono, фаворит многих дистрибутивов.**

К посылаемому по электронной почте изображению можно прикрепить заголовок, описание и тэги. Название фотографии вставляется в тему письма, а описание – в качестве основного текста. Тэги задаются в любом из этих полей при помощи выражения «tags: имена_тэгов». Вы можете послать фотографию с вашего Linux-компьютера на учетную запись Flickr, отправив такое письмо:

Кому: вашеимя+family@photos .flickr.com
Тема: Мой новенький компьютер tags: Компьютер Linux Рождество
Текст: Глазам не верю: муж подарил мне на Рождество новенький компьютер с Linux!

Вложите в письмо фотографию и отправьте его. Через некоторое время это фото попадет в ваш альбом.

С рабочего стола

Еще один удобный и платформо-независимый способ загрузки изображений в Интернет – воспользоваться *Uploadr*, фирменным инструментом Flickr. Он похож на мини-программы для прикрепления файлов к письмам, которыми пользуются при доступе к почтовому ящику через сетевой интерфейс, и очень прост в работе. Однако более элегантное решение – отправка фотографий из программы-каталогизатора, которую вы используете локально на своем компьютере. Важно, чтобы программа была универсальной, то есть позволяла бы списывать снимки с фотокамеры, редактировать их и затем пересылать в online-службу. Лучшими такими программами для Linux являются *F-Spot* и *DigiKam*. Первая основана на Mono и поэтому имеет много зависимостей, но зато входит в стандартный набор многих дистрибутивов. *DigiKam* – стабильная часть KDE и по идее доступна всюду.

Обе программы имеют свои сильные стороны (*см. Сравнение в LXF73*), но меня интересуют их возможности по комментированию снимков и загрузки их на *Flickr*. Начнем с *F-Spot*, где интеграция с *Flickr* была реализована с самых ранних версий. Изображениям в этой программе можно присвоить и свои тэги, и заранее созданные шаблоны, и в соответствии с ними упорядочить. Приятнее всего, что эти тэги понятны *Flickr* без редактирования, поэтому подписи и комментарии F-Spot можно напрямую экспортировать в сетевую службу. В *F-Spot* поддерживается распространенный способ выделения нескольких изображений: с помощью Shift – последовательно расположенных снимков, а с помощью Ctrl – произвольных. Выбрав нужные фотографии, воспользуйтесь опцией File > Export > Export to Flickr. Появится диалоговое окно, где можно настроить некоторые параметры перед собственно отправкой. Здесь вы можете при желании удалить метаданные снимков, включая фокусное расстояние при съемке, дату, время и прочее, а также изменить размер фотографий и указать, надо ли конвертировать тэги *F-Spot* в формат *Flickr*. Если ваши тэги не вписываются в стандартные рубрики *Flickr*, будет созда-

КАК НАСЧЕТ СЕТЕВЫХ ОФИСНЫХ ПРИЛОЖЕНИЙ?

Офисные приложения, доступные через web-интерфейс, были на подходе еще до того, как компания Sun начала раскручивать Java. Конечно, в бизнесе ценится любой способ снизить затраты на IT, оптимизировать работу компьютеров и повысить при этом безопасность и управляемость. Но эта волокита тянется еще с девяностых, хотя при Web 2.0 больше разглагольствуют об Ajax (Асинхронный JavaScript и XML), чем о Java. Много говорилось и о тратах Google на перевод *OpenOffice.org* целиком в web.

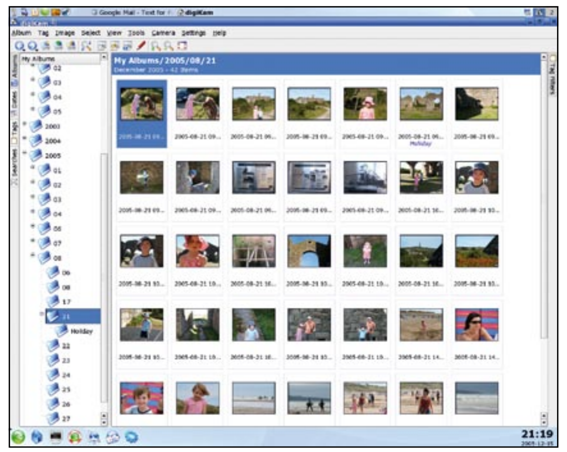
Уже существует целый спектр программ, основанных на Ajax, растет число и web-приложений. Так, Gmail можно назвать эффективным текстовым процессором начального уровня. Тут есть набор средств форматирования текста, инструменты выделения и автоматической вставки web-ссылок, а также проверка орфографии. Gmail прекрасно работает в *Firefox* (и любом другом браузере) на любой платформе. Недавно программа обзавелась функцией автосохранения: черновик текущего документа каждые пару минут в фоновом режиме записывается в специальную папку. Закончив работу над документом, вы можете мгновенно послать его адресату (очень удобно для совместной работы) или распечатать прямо из почтовой программы. При работе с другими людьми черновики также автоматически помещаются в папку Sent (Отправленные), поэтому вы никогда не потеряете ценный документ, поскольку каждый получает метку и легко разыскивается через интерфейс Gmail.

Так что, пока мы следили за развитием *OpenOffice.org*, Google успела выпустить классный, кроссплатформный текстовой редактор с встроенным автосохранением и почтовыми функциями. Помолимся, чтобы в нем скорее появился подсчет слов…

на новая рубрика. После этого остается лишь ввести ваше имя пользователя и нажать кнопку «Export».

Теперь займемся *DigiKam*, в настоящее время это самый функциональный фото-менеджер для KDE. Благодаря использованию расширений *Kipi*, программа неплохо справляется с экспортом в *Flickr*. Но, в отличие от *F-Spot*, программа не пытается использовать свою систему тэгов при работе с web-службами. Это означает, что в отдельных случаях вам придется присваивать тэги к одной и той же фотографии дважды. Однако этот недостаток искупается возможностью задавать правила доступа к загружаемым во *Flickr* фотографиям.

ЧАСТЬ 3 – ЗАГРУЗКА ЧЕРЕЗ DIGIKAM

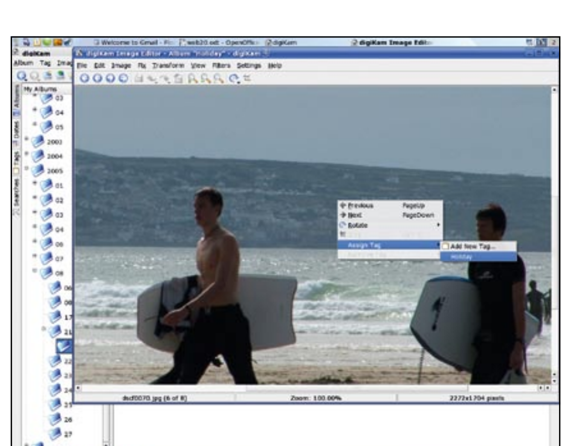


Интерфейс DigiKam

Для тех, кто склонен использовать родные KDE-приложения, *DigiKam* предлагает лучшую комбинацию формы и содержания, и хотя он выглядит немного сложнее *F-Spot*, интеграция с *Flickr* здесь проста и понятна. Очевидно, конечно, что все функции *Flickr* обеспечиваются работой дополнительного расширения, а не собственным инструментарием программы, и поэтому у вас не получится выделить несколько изображений в главном окне *DigiKam* и тут же скинуть их в Интернет.

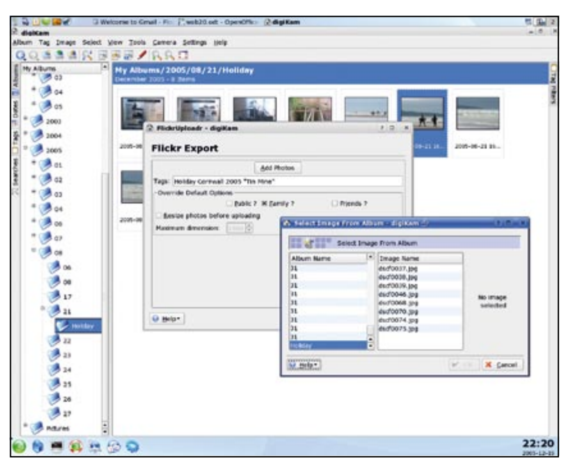
Собираем фотографии

Учитывая особенности экспорта файлов в *DigiKam*, для пересылки в *Flickr* проще всего выделить нужные фотографии и поместить их в новый альбом. Выберите Album > New Album и задайте альбому имя. Если вы уж очень пунктуальны, можете завести специальный альбом для «транзитных» снимков. Перетащите в созданный альбом фотографии обычным способом (копированием, а не перемещением). Потом выделите альбом.



Редактируем и размечаем

При открытии альбома с фотографиями *DigiKam* предлагает набор эффектов и инструментов редактирования. В режим редактирования изображение переходит, если дважды щелкнуть на нем в окне браузера; тут его можно подправить при помощи инструментов из меню. Чтобы присвоить изображению специфичный тэг *DigiKam*, щелкните по нему правой кнопкой мыши и выберите пункт Assign Tag > Add New Tag (или выберите готовый тэг). Здесь же можно изменить размер фотографии – я убедился, что так быстрее, чем через диалог экспорта: воспользуйтесь пунктом меню Transform > Resize и выберите требуемый размер изображения (включив флажок, отвечающий за пропорциональность масштабирования).



Контрольная проверка

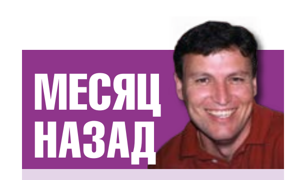
Обеспечив все настройки, возвратитесь в браузер и, выделив альбом, выберите пункт меню Album > Export > Export To Flickr. Не забудьте, что задать тэги, настроить права доступа и отмасштабировать снимки нужно до их выделения. Теперь нажмите кнопку Add Photos и выделите все содержимое вашего альбома. Загрузка начнется после того, как вы нажмете OK; поэтому не спешите, проверьте все еще раз. Если вы впервые занялись загрузкой фотографий, то появится окно *Konqueror* с предложением создать свою учетную запись на *Flickr*. LXF



РЕДАКТИРОВАНИЕ ИЛЛЮСТРАЦИЙ И ИЗОБРАЖЕНИЙ

Gimp Световые эффекты

«Моментальные маски» и слои **Майкла Хэммела** (Michael J Hammel) отучат ваших детей соваться на родительские вечеринки.



Работая с тенями и перспективой, я показал, как можно создать объемное изображение упаковочной коробки.

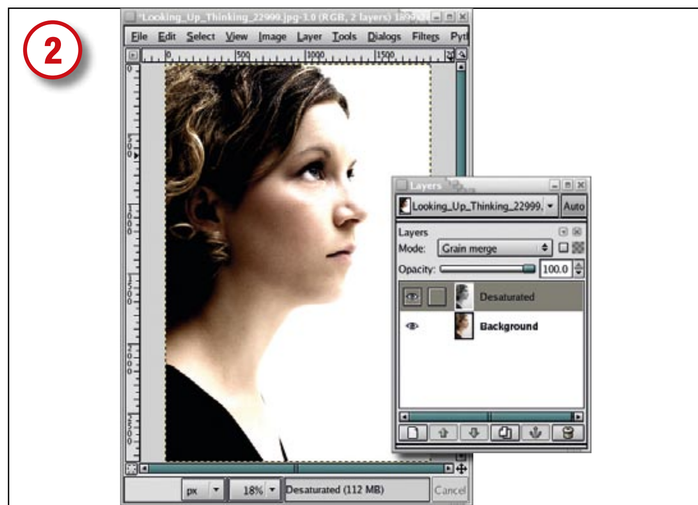


Я люблю кино. Кино – трогательно. Оно заставляет нас улыбаться, смеяться и плакать. Оно поднимает нас к таким эмоциональным вершинам, каких ни за что не достичь телевидению. Интересно, что точно такое же воздействие на нас оказывают наши подросшие дети (если вы сами, э-э, ребенок-подросток – теперь вы знаете, откуда берутся лучшие родительские чувства). Поэтому лучшей темы для нашего нынешнего урока по *Gimp* не придумаешь: выполним в виде киноафиши объявление о родительской вечеринке в местной средней школе.

Стиль афиши годится для анонса любого события, будь то школьный спектакль, пятничная тусовка в ближайшем баре или даже распродажа старья соседской семьей! Броская, впечатляющая афиша никого не оставит равнодушным, а тот, кому вы ее предназначили, просто не сможет пройти мимо. Киноафиша начинает рассказ, но не выдает, чем он кончится, она призывает к соучастию – а вам, конечно, подавай побольше соучастников, будь то обсуждение новейшего техно-триллера или угощение фирменным блюдом. В процессе урока мы с вами привнесем в нашу афишу элемент научной фантастики, символизирующий Первый Контакт между двумя мирами – родителей и школьников. Главная наша задача – добиться эффекта неземного свечения лиц школьников, свидетельствующего о встрече с Высшими Силами. Информацию о самом событии поместим внизу, наподобие титров фильма.

Для воспроизведения эффекта золотистого сияния свыше *Gimp* предусмотрел технику «моментальных масок», инструмент выборки контура и цветные слои. Поможет также галерея изображений. Постоянные читатели, наверное, заметили, что мой любимый сайт с картинками – **BigStockPhoto.com**. Отсюда можно загрузить отличное фото, с высоким разрешением, всего за $1. Я взял за основу фотографию юной девушки, с печатным размером 1900x2850 пикселей и разрешением 300 dpi. (Чтобы добыть именно эту фотографию, зайдите на страницу Photo Search сайта **BigStockPhoto.com** и возьмите картинку ID 22999.)



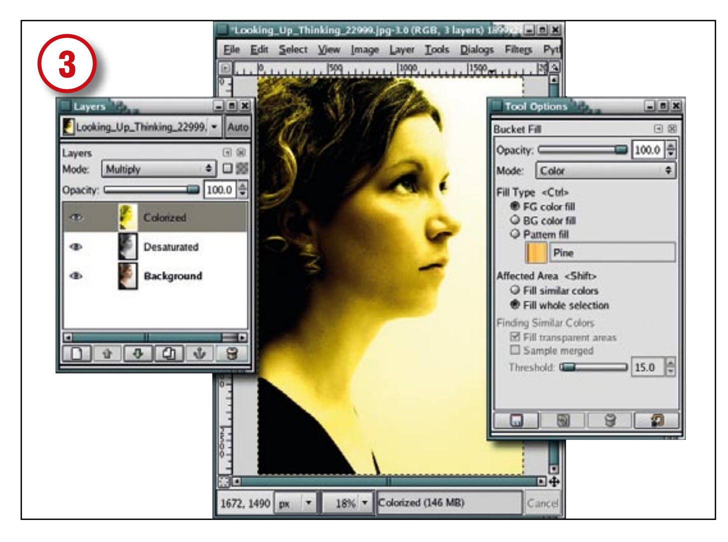


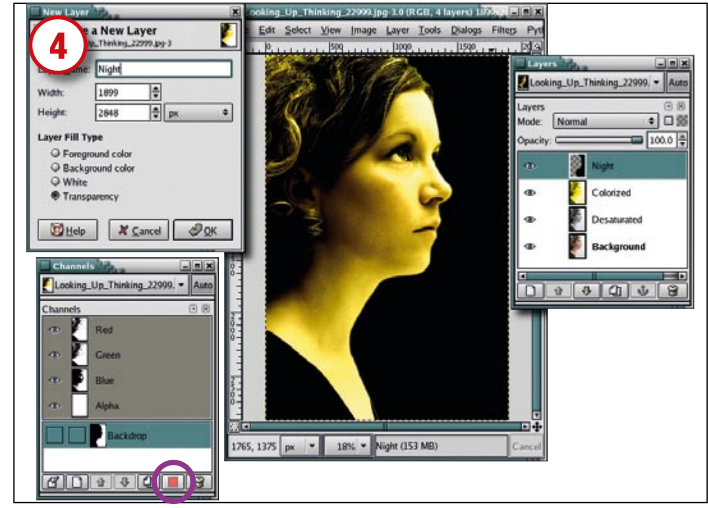
Отделение фона

Для начала поменяем цвет фона на черный: инструментом Fuzzy Select (Создание контура) отделим светлую часть изображения. Так как светлый фон практически однороден, инструмент легко, за пару щелчков мыши, выделит светлое пространство, лишь слегка прихватив лицо и декольте. Выборка конвертируется в «моментальную маску» с помощью кнопки в левом нижнем углу холста (на правом фото обведена кружком). «Моментальная маска» позволяет подчистить выборку мягкой кистью, черной или белой. Нажав ту же кнопку, вернитесь с маски на выборку, затем растушуйте выборку (Select > Feather, Выделение > Растушевка) на 10 пикселей. В предыдущих уроках я задавал гораздо меньшую растушевку, но сейчас мы работаем с размером печатного плаката.

Ослабление яркости

Теперь убавим красок в лице девушки. Сохраните выборку в канале (Select > Save To Channel, Выделение > Сохранить в канале), дав ей имя Backdrop (Запас). В окне холста отключите выборку, нажав Ctrl+Shift+A. Сохранение выборки в канале передает активность следующему каналу, поэтому щелкните на исходном слое в диалоговом окне слоев, чтобы снова его активировать. Дублируйте слой (Layer > Duplicate Layer, Слой > Создать копию слоя) и назовите его Desaturated (Бледный). Осветлите новый слой (Layers > Color-Desaturate, Слой > Осветление) и установите значение его режима Grain Merge (Убрать зернистость). Контрастность изображения усилится, чего и следует ожидать при свете, падающем на лицо девушки в ночи.





Добавим золотистый слой

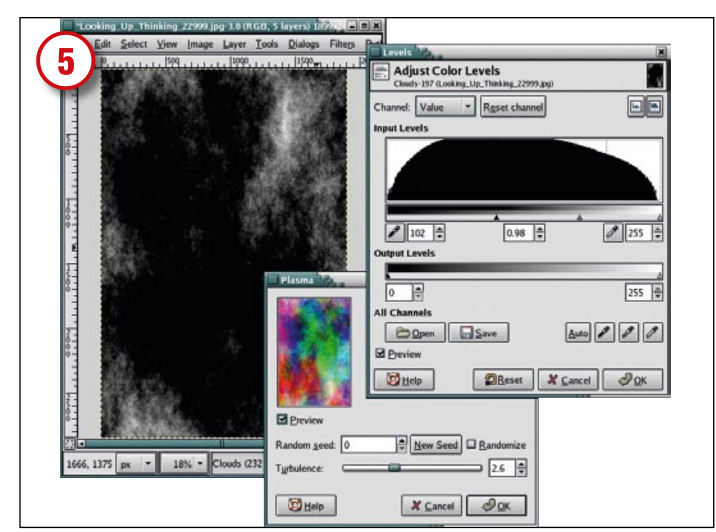
Дублируйте Бледный слой и назовите дубликат Colorized (Цветной). Двойной щелчок в окне выбора цвета переднего плана (Foreground Color) на Панели инструментов откроет диалоговое окно. В текстовом поле введите значение FCFF00 (это золотисто-желтый цвет). Чтобы изменения вступили в силу, нажмите OK. Выберите в Панели инструментов Bucket Fill (Заливка), и в диалоговом окне этого инструмента установите режим «Цвет». Выделите весь слой (Select > All, Выделение > Выделить все), затем щелкните мышью на холсте для окраски слоя. Убедитесь, что режим данного слоя установлен как Multiply (Умножение). Это придаст лицу девушки золотистый оттенок и немного затенит изображение.

Затемнение фона

Активизируйте слой Цветной слой в диалоговом окне слоев, щелкнув по нему. Добавьте прозрачный слой (Layer > New, Слой > Новый слой) поверх стопки и назовите его Night (Ночь). Теперь вернем выборку из канала Запас, открыв диалоговое окно каналов (Dialogs > Channels, Диалоговые окна > Каналы), выделив нужный канал и нажав кнопку «Канал» справа внизу окна (на рисунке обведена кружком). Выделите слой «Ночь» в диалоговом окне слоев, чтобы сделать его активным. Увеличьте выборку на 3 пикселя (Выделение > Увеличить) и растушуйте ее (снова) на 10 пикселей. В окне изображения введите D, то есть назначьте черный цвет для фона, затем перетащите передний план на выборку.









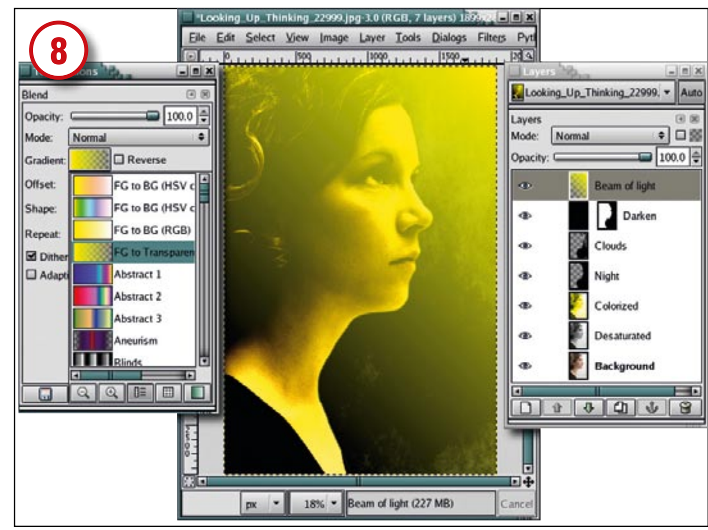
Эффект дымки

Просто черный фон – это скучно, давайте немножко его задымим. Вот как это делается: добавим прозрачный слой под названием Clouds (Облака). Откроем фильтр Плазма (Filters > Render >Clouds > Plasma, Фильтры > Визуализация > Облака > Плазма), зададим турбулентность 2.6 и нажмем OK, чтобы применить изменения. Обесцветим слой (Layers > Color > Desaturate, Слой > Цвета > Обесцветить) для уменьшения количества видимого дыма. Дымка нужна в основном в правом верхнем углу изображения, и для ее перемещения нам может понадобиться инструмент Flip (Подтолкнуть).

Уберем лишний дым

Снова вызовите слой Запас. Убедитесь, что после его вызова слой Облака в диалоговом окне слоев выделен и активен. Инвертируйте выделение (Select > Invert, Выделение > Инвертировать) и растушуйте его на 10 пикселей, затем, нажав в холсте Ctrl+X, вырежьте выборку из облачного слоя. Если дымка чересчур яркая – уменьшите яркость слоя Облака, как показано на рисунке.



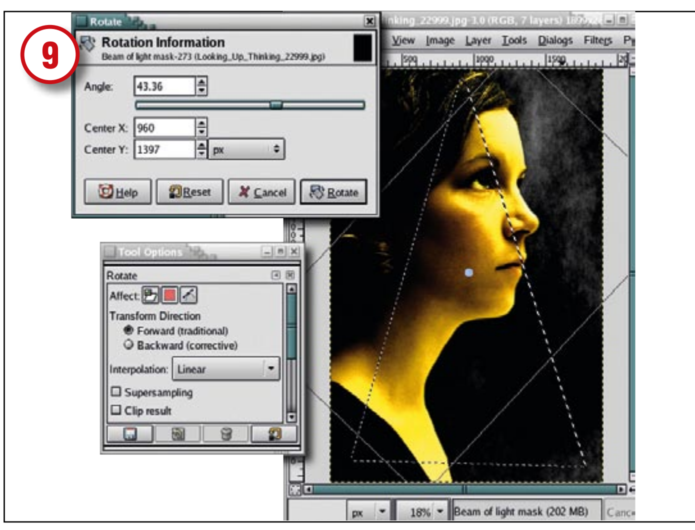


Затемнение лица

Хотя золотистый слой уже несколько затемнил лицо девушки, оно все еще светловато. Затемним его еще немного: добавим новый слой. Снова наберите D на холсте, чтобы установить цвет переднего плана в черный. Создав новый слой (Слой > Новый слой) и назвав его Darken (Тень), залейте его черным, перетащив на холст цвет переднего плана из Панели инструментов. Добавьте белую маску слоя (Слои > Маски > Добавить маску слоя). Еще раз вызовите сохраненную выборку Запас и растушуйте ее на 10 пикселей. Активируйте маску слоя Тень, выделив ее, затем залейте выборку черным. Задайте значение режима смешения слоев для слоя Тень как Overlay (Перекрытие) и очистите выборку (Ctrl+Shift+A).

Источник света

Настала очередь освещения. Для начала добавьте новый прозрачный слой сверху стопки и назовите его Beam of light (Луч света). Щелкните мышью в окне выбора цвета переднего плана и введите в текстовом поле значение FCFF00, мы уже использовали этот золотистый цвет. Нажмите OK, чтобы применить значение цвета. Теперь выберите инструмент Blend (Смешение) в Панели инструментов, и в диалоговом окне настройки инструмента установите значение градиента переднего плана как Transparent (Прозрачный). Проведите линию из правого верхнего угла холста через глаза девушки до тыльной стороны ее шеи. Теперь справа исходит желтое свечение, пока что не сфокусированное в луч.

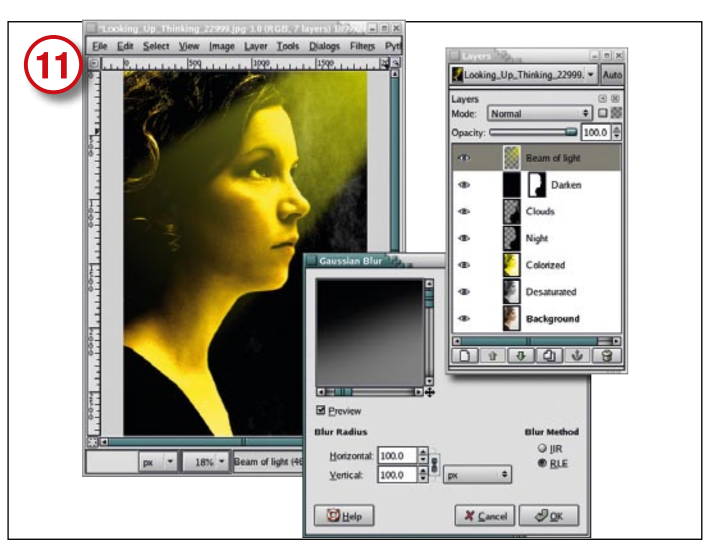


Настройка светового луча

Добавьте черную маску к слою Луч света. Инструментом рисования контура из Панели инструментов создайте треугольник с тремя контрольными точками. Затем в окне настройки параметров инструмента конвертируйте контур в выборку, щелкнув Create Selection From Path (Создать выборку по контуру). В Панели инструментов выберите Rotate (Вращать), и в окне настройки параметров под заголовком Affect (Воздействовать на) нажмите кнопку Transform Selection (Преобразовать выборку), среднюю из трех кнопок вверху этого диалогового окна. Щелкните мышью на холсте и используйте ползунок настройки значения угла поворота в информационном окне для поворота контура выборки по часовой стрелке на угол примерно 45 градусов. Чтобы изменения вступили в силу, нажмите кнопку Вращать.

Корректировка выборки

В Панели инструментов выберите инструмент Move (Перемещение). Удерживая клавишу Alt (а может быть, заодно и Shift, в зависимости от настроек вашего рабочего стола Linux), переместите выборку вверх и вправо. С помощью инструмента Scale (Масштабирование) из Панели инструментов, перейдите в диалоговое окно настройки свойств инструмента и установите значение Воздействовать на в Преобразовать выборку; затем щелкните на холсте. Появится четырехугольная рамка области изменения масштаба. Зацепите нижнюю левую угловую точку и перетащите ее влево-вниз. Нажмите OK для применения изменений.



Направление светового луча

Работа близится к концу, но для ее завершения, при активной маске слоя «Луч света» в диалоговом окне слоев, н ажмите на холсте D и X для смены цвета переднего плана на белый. Выберите инструмент Смешение и установите значение градиента на Прозрачный передний план. Проведите линию через центр выборки от верхнего правого угла до нижнего левого, затем очистите выборку (Ctrl+Shift+A). Слегка размажьте маску (Filters > Blur > Gaussian Blur, Фильтры > Размывание > Гауссово Размывание) на 100 пикселей. Если вы считаете, что луч слишком велик, примените маску слоя (Слои > Маска > Применить маску слоя), вызовите инструменты Масштабирование или Perspective (Перспектива) и отрихтуйте луч вручную. Если он слишком яркий, приглушите яркость.



Завершение: текст

В Панели инструментов выберите инструмент Text (Текст) и измените цвет переднего плана на белый, как прежде. В окне настройки инструмента выберите подходящий шрифт (я взял XBAND Rough размером 140 пикселей для заголовка и Soutane Ultra-Light для информационной строки). Обе надписи расположите в правом нижнем углу холста. Тот же инструмент послужит для создания любых других текстовых элементов изображения. Черный фон позади белых букв сделает их хорошо читаемыми. Ну, вот и все!

Последнее замечание: когда вы уменьшите изображение с тем, чтобы увидеть его полностью, текст может исказиться. Чтобы увидеть постер в том виде, в котором он выйдет из печати, несколько раз подряд проведите операцию увеличения (View > Zoom In, Просмотр > Увеличение). LXF

ЧЕРЕЗ МЕСЯЦ

Используем градиенты для создания хоть и простого, но весьма впечатляющего эффекта зеркального стекла.

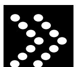
3D-МОДЕЛИРОВАНИЕ

Realsoft 3D Первые шаги

Следуйте за **Ником Вейтчем**: нарисуем несложный 3D-объект.



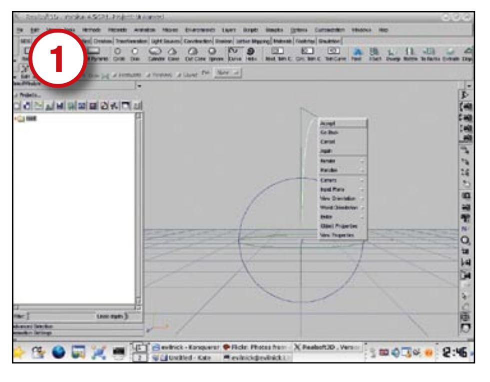
В HTML-руководстве для Realsoft 3D много советов и разобранных примеров.



На заре развития 3D-приложений инструменты были попроще теперешних – в основном потому, что они не дотягивали и до половины современной мощности. За прошедшие годы различные приложения и фирмы упражнялись во множестве методов, чтобы достичь лучших результатов в 3D. В итоге, не считая функций, все 3D приложения весьма различны в использовании. Я уверен, что *Realsoft 3D* (**http://www.realsoft.com/**, доступна пробная 60-ти дневная версия) не покажется вам трудным в освоении, однако он, возможно, не похож на программу, к которой вы привыкли, поэтому этот краткий учебник призван вам помочь. Также не помешает взглянуть на превосходное HTML-руководство, включенное в состав пакета; вы можете открыть его в любой момент, нажав последовательно пункты главного меню Help > User Manual (Подсказка > Руководство пользователя).

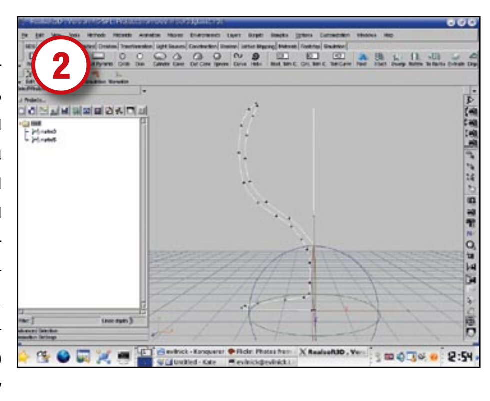


Используем Nurbs



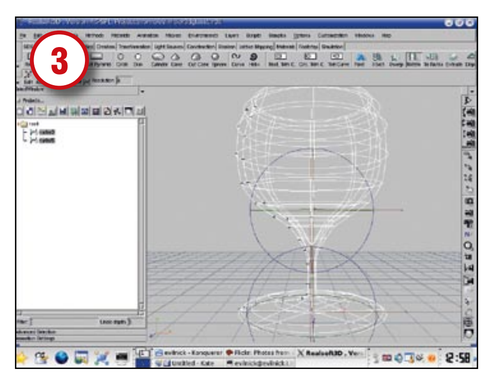
Для создания разных объектов существуют и разные инструменты. Бокал мы собираемся создать при помощи Nurbs. Это смешное слово означает «Неравномерный Рациональный Би-Сплайн» (Non-Uniform, Rational B-Spline) – это вариант математического представления 3D-объектов. Для начала нам нужна прямая вертикальная линия. Щелкните на инструменте Curve (кривая) на панели инструментов, а затем щелкните внизу посередине главного окна. Тут будет начало прямой. Щелкните вновь на точке строго над начальной для создания прямой линии – согласно Эвклиду, этого достаточно, поэтому вызовите контекстное меню правой кнопкой мыши и выберите пункт Accept (Принять).

Профиль



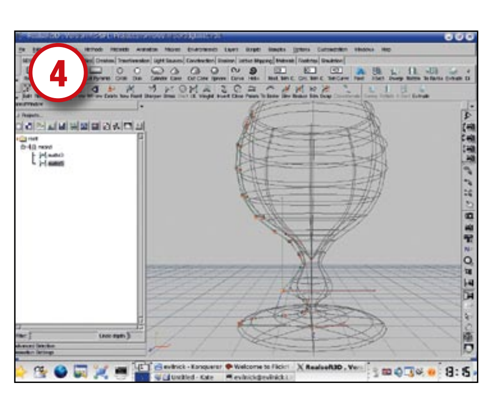
Итак, мы создали ось, вокруг которой будем строить бокал. Теперь создадим профиль бокала. Снова выберите Curve (Кривую) и нарисуйте профиль, щелкая там, где нужна узловая точка (проще будет начать снизу и продвигаться вверх). Не забудьте создать подставку, а также внутреннюю емкость. Замкнутую форму создавать не обязательно, но прежде чем щелкнуть правой кнопкой и нажать на Accept (Принять), удостоверьтесь, что контур вернулся к оси.

Вращение



Теперь у нас есть две линии: прямая и кривая, которую мы будем вращать вокруг прямой. Щелкните слева в древе объектов, чтобы выбрать ось, которую вы нарисовали. Затем, удерживая Shift, выберите профиль. Многие операции моделирования используют один или более объектов – теперь вы знаете, как это делается! Далее выберите кнопку Rotate (Вращение) около правого края панели инструментов. Вы увидите проволочный каркас, получившийся из профиля. Щелкните на кнопке Accept (Принять) на панели инструментов для сохранения модели.

Правка узлов



Одна из замечательных возможностей *Realsoft 3D* – редактирование объекта без отмены проделанной работы, если вы сочли, что ваш бокал выглядит не совсем хорошо. Слева на панели объектов в древе теперь появился сетчатый объект (Mesh object). Щелкните на '+' рядом с ним, и увидите: две нарисованные вами кривые входят в него как дочерние объекты. Щелкните на объекте-профиле и выберите пункт Edit (Редактировать) на панели инструментов – узлы объекта окрасятся в красный цвет. Щелкайте на узлах, перемещайте их – весь объект будет менять свою форму. Когда закончите, выберите Accept (Принять).

ИНТЕРФЕЙС REALSOFT 3D

Большинство пунктов этого меню, кроме File (Файл), Edit (Правка) и View (Вид), обычно доступны где-то еще.

Tools (Инструменты). Они меняются в зависимости от текущей вкладки, но здесь-то вы и выбираете создаваемые объекты.

Действия, которые обычно связаны с создаваемым объектом.

Панель объектов показывает древо текущей сцены.

Панель анимации управляет расположением текущего вида в последовательности.

Окно просмотра, в котором объект моделируется и трансформируется.

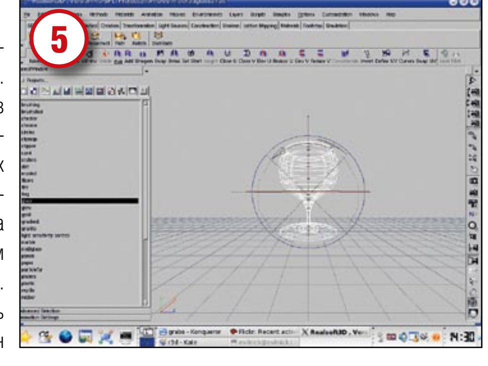
Вкладки панели инструментов обеспечивают их группировку по смыслу.

Инструменты камеры для вращения, перемещения и масштабирования вида.

Виды: спереди, сбоку, сверху. При нажатом <Ctrl> выбор вида приводит к показу с обратной стороны.

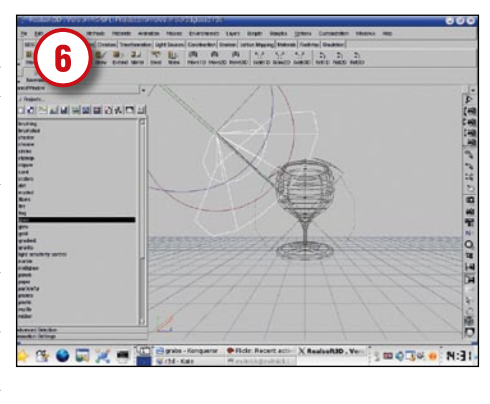
Три режима рендеринга. Сверху вниз: каркас, OpenGL и трассировка лучей.

Материалы



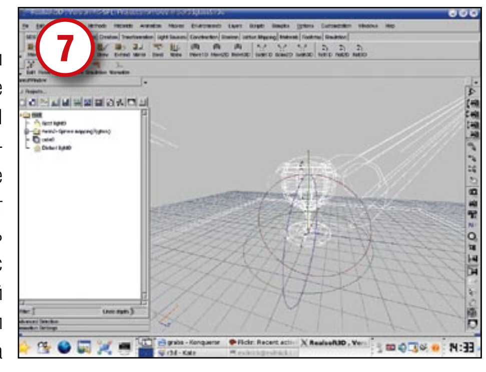
Выберем материал для объекта. Сделаем наш бокал... из чего бы? Я думаю, из стекла. *Realsoft 3D* предоставляет изобилие готовых материалов, в том числе и стекло, поэтому на этом этапе мы не станем изобретать собственный. Удостоверьтесь, что весь сетчатый объект выбран в панели объектов, затем щелкните на иконке с бело-голубой шахматкой на верху этой панели. После этого выведется список предопределенных материалов. Найдите glass (стекло) и просто перенесите его на объект в главном окне.

Подсветка



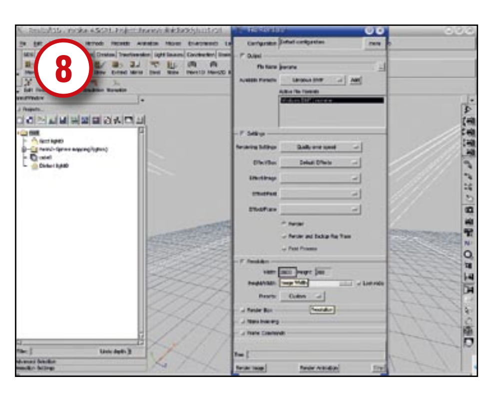
Чтобы показать наш материал в выгодном свете, потребуется... свет. Выберите Light Sources (Источники света) на панели инструментов, для доступа к инструментам освещения. Выберите инструмент spotlight (софит), затем щелкните и нарисуйте линию, показывающую направление, а также внешний и внутренний конусы луча. Постарайтесь сфокусировать их на бокале! Возможно, вам захочется добавить фоновую подсветку сбоку, иначе сцена будет темноватой. Не забывайте щелкать на кнопке Accept (Принять) после создания каждого элемента освещения. Возможно, вы также захотите выбрать другую позицию камеры, справа в основном окне, чтобы убедиться в верном расположении источников света.

Трансформация



Теперь поставим бокал на что-нибудь. Щелкните на вкладке Analitical (Аналитические) и выберите cube (куб). Перенесите его под бокал – это будет столик. Переключитесь на вид сверху (иконка с камерой, направленной вниз, справа в основном окне). Затем щелкните на пункте Transformation (Преобразование), для перемещения и установки размера созданных объектов. Выберите кнопку масштабирования (scale button) и начертите небольшой прямоугольник. Теперь подвигайте мышку в главном окне – размеры куба будут меняться. Используя комбинацию этих инструментов и положения камеры, вы сможете создать основание, где будет стоять бокал.

Снимите фото



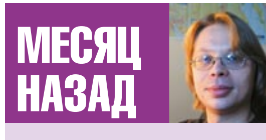
В конце этого краткого урока я покажу, как получить снимок готового изображения. Для этого надо будет установить камеру и использовать режим рендеринга. Выберите нужный ракурс в окне и щелкните на вкладке Creation (Создание). Щелкните на пункте Camera (Камера), затем Accept (Принять), чтобы создать камеру текущего вида. Для вызова диалога Render (Рендеринг), выберите в меню File > Render (Файл > Отобразить) По умолчанию создается BMP-файл Microsoft Windows, который открывается в большинстве Linux-приложений. Выберите желаемое разрешение и нажмите кнопку Render. Подождите около минуты – и готово. Добро пожаловать в мир 3D – ура! **LXF**

ВЕКТОРНАЯ ГРАФИКА

Inkscape Фигуры

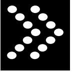


ЧАСТЬ 4 *Inkscape* обладает тем, чего не хватает в *Adobe Illustrator*: простыми в использовании, но богатыми возможностями объектами, так называемыми фигурами (shapes). **Дмитрий Кирсанов** научит вас работать с ними.



МЕСЯЦ НАЗАД

Темой прошлого занятия были контуры – ключ к работе в Inkscape.



Свобода делать то, что хочется, свобода работать так, как нравится – вот что отличает по-настоящему мощную программу. Этому определению как нельзя лучше соответствует хороший векторный редактор. Имея весьма ограниченный набор векторных типов объектов и инструментов, вы можете воспроизвести – точно или хотя бы приближенно – любое мыслимое графическое изображение.

Однако неограниченная свобода – не всегда благо. Например, объекты-контуры, которые мы изучали в прошлый раз, не знают ограничений и могут с любой степенью точности воспроизвести любую двумерную фигуру. Но зачастую это излишняя роскошь; бывает, что нужна простая четко определенная фигура. Например, прямоугольник. Конечно, его отлично можно построить как контур с четырьмя узлами, соединенными отрезками прямой – но нет ли способа попроще?

> «ИНСТРУМЕНТ «ЗВЕЗДА» – САМАЯ УВЛЕКАТЕЛЬНАЯ ВЕКТОРНАЯ ИГРУШКА ДЛЯ ВСЕХ ВОЗРАСТОВ.»

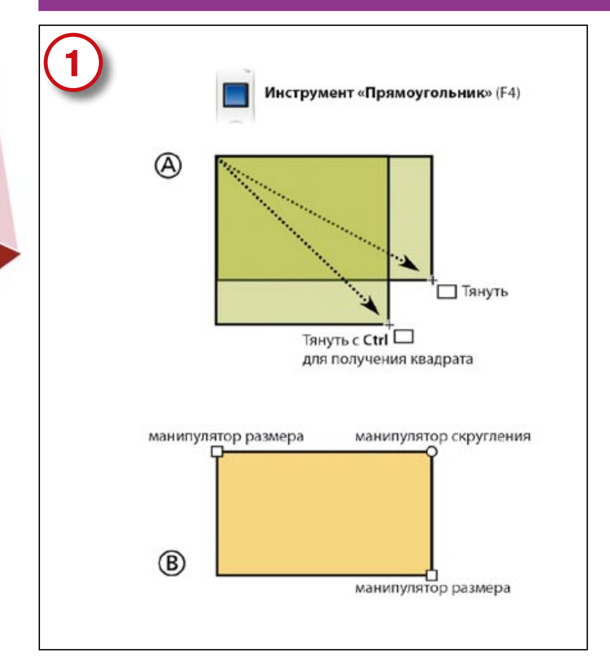
Способ есть! В *Inkscape* предусмотрено несколько типов объектов для представления наиболее распространенных геометрических фигур, и каждый из них снабжен инструментом для создания и редактирования объекта. Параметры фигур можно задавать в числовом виде либо пользоваться «рукоятками» фигур; есть и «горячие клавиши». Каждая фигура знает свой тип и предоставляет пользователю все свои, специфические для фигур данного типа возможности. С фигурами нельзя делать «все что угодно», но то, что они умеют – они делают хорошо.

Интересно, что в *Adobe Illustrator* есть инструменты для создания простых геометрических фигур. Но как только объект создан, он тотчас же забывает о своем происхождении и становится обычным контуром, без каких-либо «фигурных» способностей. Напротив, «сознательные» фигуры *Inkscape* настолько удобны, что эту программу стоит освоить, даже если вам не хочется расставаться с привычным Illustrator'ом.

На данный момент в *Inkscape* есть четыре типа фигур: прямоугольник, эллипс, звезда и спираль. Каждая фигура имеет свой рабочий инструмент на Панели инструментов слева от изображения, причем набор элементов управления инструментом зависит от типа фигуры. Действуя через эту панель или редактируя фигуры прямо в рисунке, вы можете изменять объекты в очень широких пределах – например, эллипс можно превратить в эллиптическую дугу или сегмент.

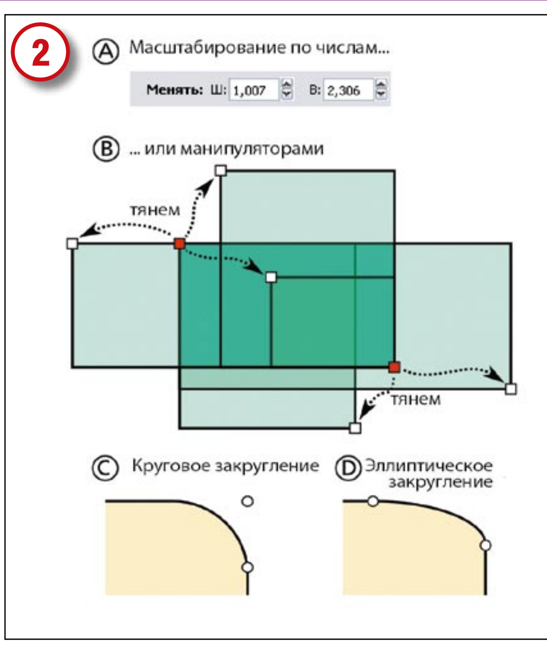
Фигуры, особенно прямоугольники и эллипсы, широко используются при создании чертежей и иллюстраций. В простых чертежах вы, вероятно, сможете обойтись вообще без контуров и кривых Безье. Но практическое удобство – это еще не все. Богатство органов управления фигурами в *Inkscape* и то удовольствие, которое они доставляют, слаженно и логично реагируя на малейшее изменение параметров, делают их интереснейшим объектом исследования для новичка. А инструмент «Звезда» – пожалуй, самая увлекательная векторная игрушка для детей всех возрастов.

АНАЛИЗ: ВОЗМОЖНОСТИ ЛИНИЙ



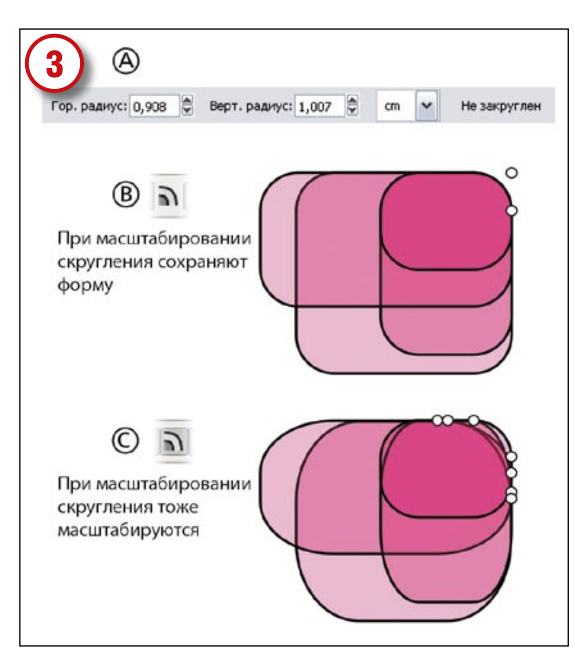
Прямоугольники: Создание

Однако начнем мы не со звезд, а с прямоугольников. При всей их простоте именно эти фигуры используются в дизайне чаще любых других. Итак, включим инструмент Rectangle (Прямоугольник), нажав F4 (потому что 4 угла – легко запомнить), и протащим мышью в любом месте холста (A). Если протаскивать при нажатой клавише Ctrl, получится квадрат. У появившегося прямоугольника есть три манипулятора (B): два квадратных (слева вверху и справа внизу) и один круглый (справа вверху).



Прямоугольники: Рукоятки

Квадратными манипуляторами можно менять размеры прямоугольника, переместив любую из сторон в любом направлении (B); пара числовых параметров (A) задает точную высоту и ширину выделенного прямоугольника. Круглый – это на самом деле два перекрывающихся манипулятора. Сдвиг одного из них от начального положения вызывает скругление всех четырех углов прямоугольника (C); если сдвинуть обе, скругление примет форму эллипса (D).



Прямоугольники: Скругление углов

Когда вы создаете несколько прямоугольников со скругленными углами, бывает нужно сохранить для всех углов один и тот же радиус скругления, даже если прямоугольники имеют разные размеры. В соответствующих числовых полях указываются горизонтальный и вертикальный радиусы скругления для любого числа выделенных прямоугольников (A). Кроме того, на панели инструмента Селектор есть специальная кнопка; если она отжата, то радиусы скругления будут сохраняться неизменными при любых трансформациях прямоугольников. Сравните B (кнопка отжата) и C (кнопка нажата).



Прямоугольники: Трансформации

Возможно, вы недоумеваете: зачем в *Inkscape* нужны органы управления шириной и высотой прямоугольника? Ведь прямоугольник (как и любой другой объект) можно растягивать и сжимать Селектором… Дело в том, что Селектор всегда трансформирует объект, пропорционально изменяя размеры сторон, что искажает углы (A). Собственные же манипуляторы прямоугольника позволяют всегда сохранять исходное направление «вертикали» и «горизонтали» фигуры, как бы прямоугольник ни был искажен или растянут.

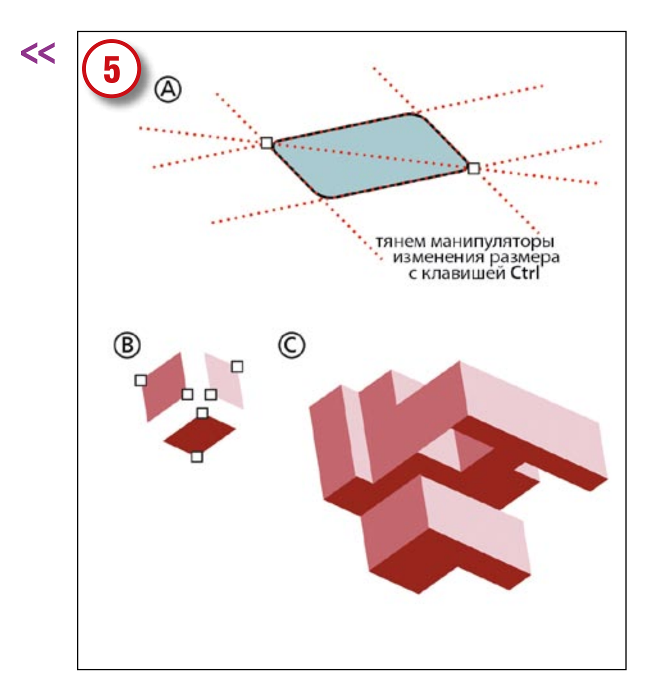
ПОДСКАЗКИ



- Рукоятки выделенных фигур можно перетаскивать и в инструментах рисования фигур, и в инструменте Node (Узел).
- Инструменты фигур запоминают заданные вами параметры и применяют их ко всем заново создаваемым объектам. Чтобы сбросить эти настройки, воспользуйтесь кнопкой Defaults (По умолчанию) соответствующего инструмента на его Панели управления.
- По умолчанию, все вновь созданные фигуры (за исключением спиралей) получают последний присвоенный какому-либо объекту стиль (например, если вы покрасили что-то в красный цвет, то и созданная после этого фигура тоже будет красной). *Inkscape* можно отучить от этого, открыв диалоговое окно Preferences (Параметры) (Ctrl+Shift+P) и жестко задав стиль заново создаваемых объектов для тех или иных инструментов.
- Поскольку объекты *Inkscape* по своим возможностям значительно богаче фигур, предписанных стандартом SVG, для построения своих фигур *Inkscape* использует элементы <path> вместо <ellipse>,<polygon> и т.д. (хотя элементы SVG тоже читаются).

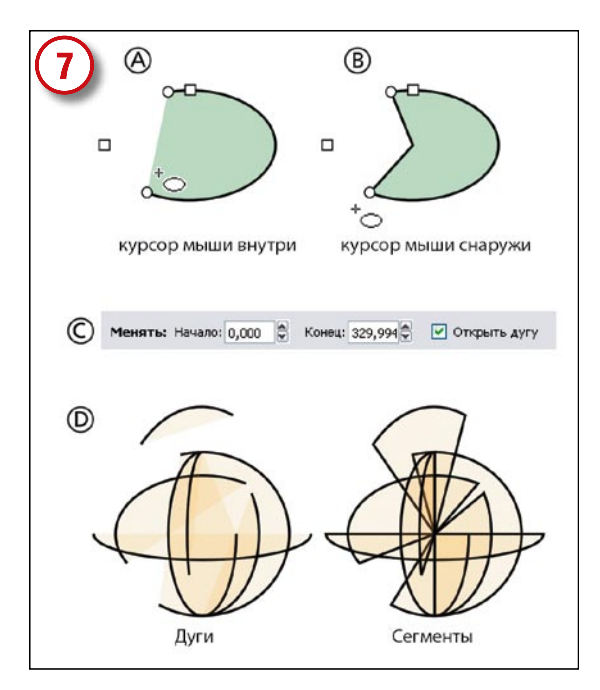
>>

<<



Прямоугольники: Прилипание

Перетаскивание квадратных рукояток прямоугольников при нажатой клавише Ctrl заставляет их прилипать к прямым, на которых лежат стороны, или же к диагонали прямоугольника (A). Это свойство позволяет рисовать несложные объемные сцены: создайте три исходных параллелограмма (B), затем дублируйте их и изменяйте размер и положение копий перетаскиванием манипуляторов с Ctrl. Рисунок (C) создан только инструментом Прямоугольник и командой дублирования (Ctrl+D); Селектор не понадобился.

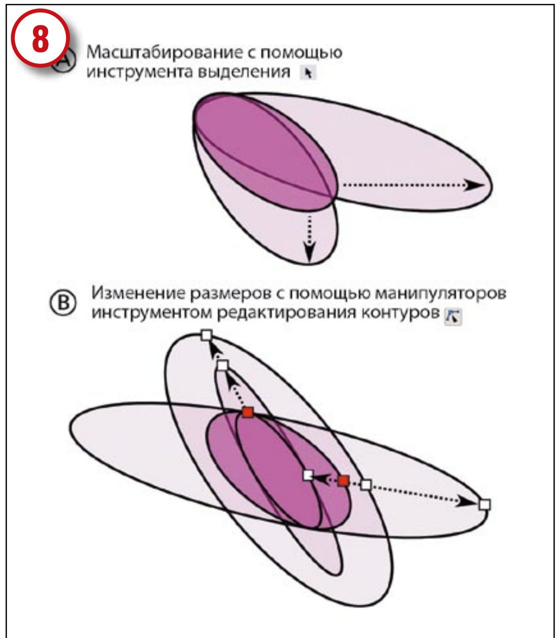


Эллипсы: Дуги и сегменты

Два круглых манипулятора эллипса выполняют иные функции, чем в прямоугольнике. Перемещая их, можно превратить эллипс в открытую дугу (A) или в закрытый сектор (B), в зависимости от положения курсора мыши (внутри или снаружи эллипса). Перемещение манипуляторов дуги или сегмента при нажатой клавише Ctrl изменяет угол ступенчато, по 15 градусов. С помощью числовых полей (C) можно точно задать начальный и конечный углы дуги или сегмента. Флажок Open Arc (Открыть дугу) превращает объект из дуги в сегмент и обратно (D).

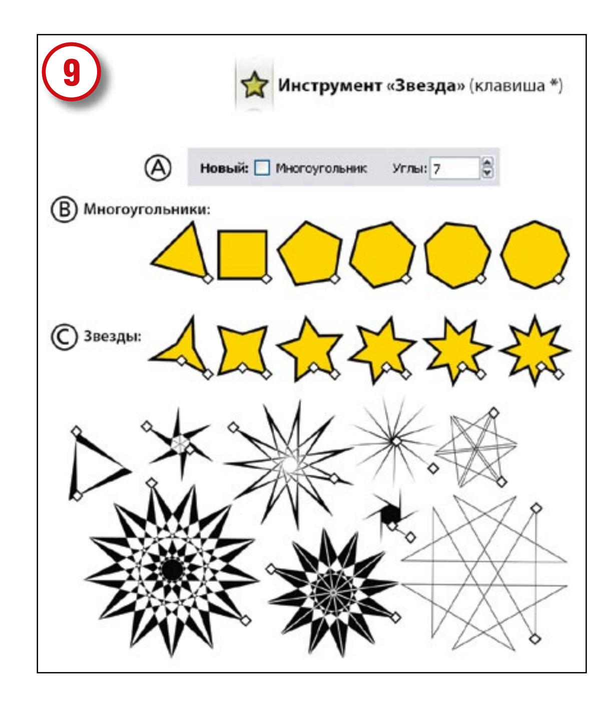
Эллипсы: Создание

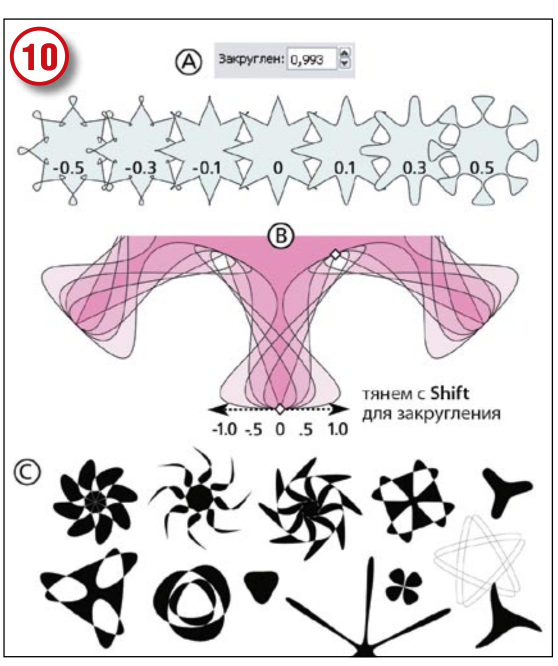
Создать эллипс не сложнее, чем прямоугольник: переключитесь в инструмент Эллипс клавишей F5 и протащите мышью по холсту (A). Если при этом нажата клавиша Shift, то точка начала перетаскивания станет центром; в противном случае эллипс впишется в прямоугольник, образованный вашим перетаскиванием. При нажатой клавише Ctrl получится правильный круг. Как и у прямоугольника, у эллипса есть три манипулятора, два из которых (квадратные) служат для изменения размеров по двум координатам (B).



Эллипсы: Трансформации

Квадратные манипуляторы эллипса работают почти так же, как у прямоугольника: как бы ни был повернут или перекошен эллипс, они помнят исходное положение его осей и позволяют растягивать и сжимать эллипс вдоль этих «внутренних» осей, сохраняя центр неподвижным (B). (Как вы помните, инструмент Селектор может растягивать или сжимать объект по горизонтали и вертикали только в экранной системе координат (A).)



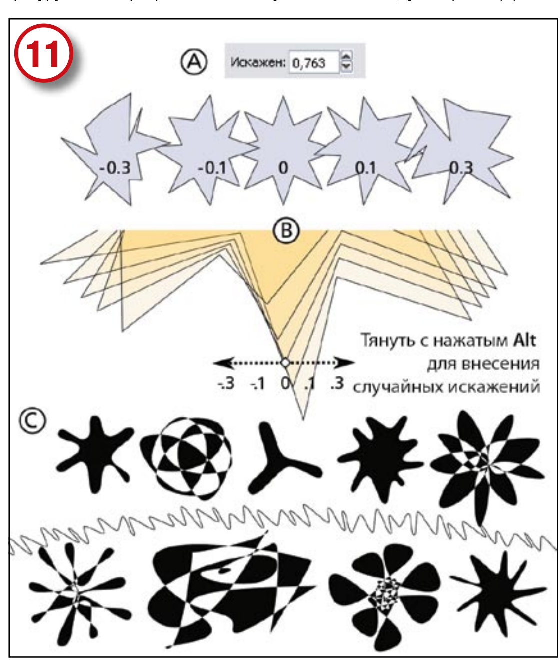


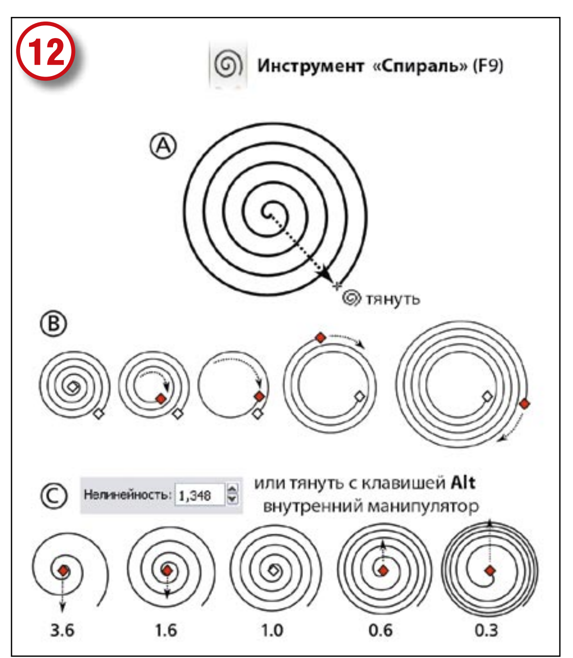
Звезды: Создание

Инструмент Звезда (клавиша *) умеет создавать два типа объектов: правильные многоугольники и звезды. У многоугольника есть только один ромбовидный манипулятор (B), управляющий его размером. Звезда с ее двумя манипуляторами – гораздо более интересный объект: их перетаскиванием можно создавать огромное разнообразие фигур, симметричных относительно поворота (C). На панели управления инструмента можно задать количество углов многоугольника или лучей звезды; кроме того, фигуру можно превратить из многоугольника в звезду и обратно (A).

Звезды: Закругление

Как и прямоугольник, звезду можно скруглить. При этом лучи ее теряют остроту, а боковые стороны из прямых превращаются в элегантные кривые Безье. Сделать это можно, изменяя параметр закругления (Rounding, A) или вращая звезду за один из манипуляторов при нажатой клавише Shift (B). Отрицательные или слишком большие положительные значения этого параметра приводят к появлению петель и самопересечений. Комбинируя закручивание с перемещением манипуляторов, вы можете создавать фигуры самых невообразимых форм (C).





Звезды: Случайность

Другой способ разнообразить звезду – внести элемент случайности. Это действие смещает все точки звезды в случайных направлениях, причем величина смещения контролируется соответствующим числовым параметром (Randomized, A). При вводе отрицательных значений направление смещения точек меняется на противоположное. Для внесения элемента случайности можно также перетаскивать манипулятор звезды с нажатой клавишей Alt (B). Если звезда закруглена, плавность закругления всех ее лучей (C) сохраняется.

Спирали

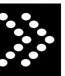
Инструмент Спираль (Spiral) (F9) рисует спираль, начиная с центра. У созданной спирали появляются два манипулятора по обоим концам (A), при помощи которых можно скручивать или раскручивать спираль (B), то есть делать ее длиннее или короче изнутри или снаружи. У спиралей есть также параметр *нелинейности* (сходимости): если он равен единице, то витки спирали идут равномерно; если больше единицы, то расстояние между витками постепенно уменьшается к центру; если меньше единицы, то наоборот. Сходимость можно менять либо числовым параметром на панели, либо перетаскиванием внутреннего манипулятора спирали при нажатой клавише Alt (C). LXF



ТЕКСТОВЫЙ ПРОЦЕССОР

Emacs для пользователя

ЧАСТЬ 1 И вновь продолжается бой! **Александр Кузьменков** подливает масла в огонь древнего противостояния Vi и не-Vi.



Emacs – пожалуй, один из самых известных проектов-долгожителей, разрабатываемых под лицензией GPL. Когда-то давно, когда деревья были маленькими, а компьютеры – большими, небезызвестный Ричард Столлмен (Richard Stollmen) начал разрабатывать свою свободную Unix-подобную операционную систему GNU (Gnu's Not Unix). Ричард относительно легко справился с написанием комплекта компиляторов GCC и множества других программ, среди которых был и *Emacs*, который задумывался им как текстовый редактор для программирования. И хотя своей главной цели Столлман пока не достиг (проекту ОС GNU не хватает стабильного ядра: работа над Hurd ведется довольно вяло), его GCC и Emacs приобрели такую популярность, что были портированы на большинство распространенных операционных систем и платформ.

Наверняка вы уже наслышаны о «святых войнах юниксоидов» на тему «*Vim против Emacs*», и скорее всего пробовали запускать *Emacs*. А может быть, вы – приверженец *Vim* и решили посмотреть на вечного конкурента своего любимого редактора?. Скорее всего потыкавшись в кнопки, расположенные на панели инструментов, вы побыстрее закрыли его, чтобы с чистой совестью вернуться в любимый *Kate*/*Gedit*/*Evim*/*Tea*, где все просто, логично и знакомо. Да, *Emacs* никак не подходит под определение «дружественной к пользователю программы» и вряд ли вам понравится с первого взгляда, но поработав с ним недельку-другую и «заточив» под свои нужды, вы вряд ли будете продолжать использовать *Kate*/*Gedit* для правки текста, а *KDevelop*/*Anjuta*/*Eclipse* – в качестве IDE. Подлинная мощь Emacs состоит в расширяемости: с помощью дополнительных компонентов он может заменить и почтовый клиент, и web-браузер, и ICQ/Jabber/IRC-клиент, и даже MP3-плеер.

Ядро *Emacs*, написанное на C, по сути, представляет собой интерпретатор языка Elisp (Emacs Lisp), а остальная часть редактора написана на этом языке. К чему такие сложности? На самом деле все довольно просто: *Emacs* – поистине огромный проект, и такой подход к разработке сделал его портирование на другие платформы и ОС довольно несложным делом. То есть оно сводится к портированию ядра программы – интерпретатора Elisp.

Стоит упомянуть, что существует несколько различных версий *Emacs*: это GNU *Emacs*, до сих пор поддерживаемый Ричардом Столлменом; отпочковавшийся от него *XEmacs*; *microEmacs*, по слухам, являющийся любимым редактором Линуса Торвальдса, и еще несколько других. Мы будем ориентироваться на GNU *Emacs*, как на самую распространенную версию, поскольку она есть во всех известных дистрибутивах, в отличие от тех же *XEmacs* и *microEmacs*.

Основы основ

Давайте разберемся с тем, как работает этот монстр. Запустите *Emacs* и нажмите **C-x C-f**. В терминологии *Emacs* **C** означает клавишу Ctrl, а **M** – клавишу Meta, соответствующую клавишам Alt или Esc. То есть «**нажмите C-x C-f**» означает, что вы должны нажать x при зажатом Ctrl, а затем, не отпуская Ctrl, нажать f. Эта комбинация используется для открытия и создания файлов, а для сохранения служит **C-x C-s**. Да, вот такие неудобные на первый взгляд «горячие клавиши» выбраны в Emacs для таких простых действий. Впрочем, стоит сказать, что каждый опытный «емаксер» создает свою собственную раскладку клавиатурных комбинаций. При нажатии **C-x C-f** в нижней части окна, называемой

СОБЕРИ САМ

Эта врезка специально для неутомимых экспериментаторов, которые всегда хотят иметь самые новые версии программ, и при этом готовы мириться с возможными ошибками.

Среди пользователей *Vim* широко распространено мнение о том, что, дескать, разработка *Emacs* ведется слишком медленно, поэтому новые версии выпускаются слишком редко. На самом же деле *Emacs* обновляется каждый день – достаточно посмотреть на CVS.

Emacs с каждым новым релизом содержит все больше и больше модулей: разработчики следят за тем, какие модули пользуются наибольшей популярностью, и стараются включить их в следующую стабильную версию редактора. Да и сами создатели модулей часто проявляют инициативу. Так, например, в 22-ю версию, которая сейчас активно развивается, попали популярные модули **ido** (*см. врезку*) и **cua**, позволяющий использовать стандартные для Windows/MacOS-программ горячие клавиши (кстати, **ido** и **cua** – творения одного человека – Kim F. Storm, см. **http://www.cua.dk**).

В новой версии улучшена поддержка *Unicode*, появилась новая документация (в том числе и на русском языке). Кроме того, теперь для сборки редактора можно использовать GTK2, что, безусловно, порадует ценителей прекрасного.

Итак, если вы хотите уже сейчас воспользоваться всеми благами *Emacs 22*, то приготовьтесь к трудностям: рабочие снимки CVS-дерева регулярно никто не выкладывает. А поскольку редактор обновляется каждый день, невозможно точно сказать, соберется у вас сегодняшняя версия или нет.

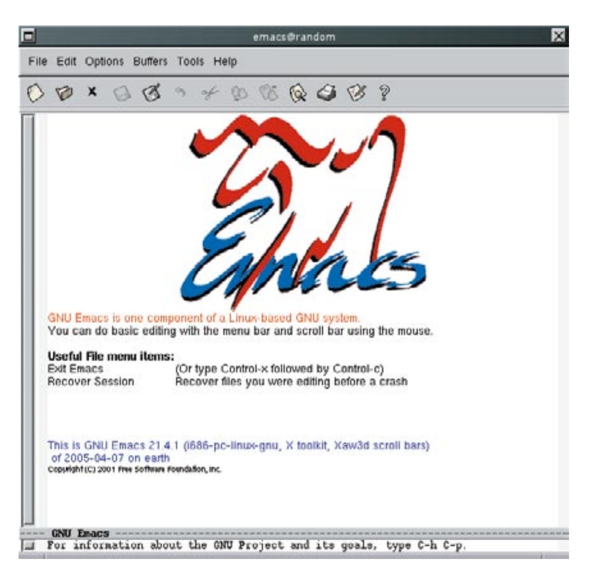
Если вас это не пугает, то можно приступить к получению CVS-дерева:

```
export CVS_RSH=»ssh»
cvs -z3 -d:pserver:anonymous@cvs.savannah.gnu.org:/sources/emacs co emacs
```

После завершения этой довольно длительной операции (еще бы, исходные тексты *Emacs* со всеми включенными модулями занимают около 90 Мб!) можно приступить к сборке:

```
cd emacs
./configure --prefix=/usr --with-xpm --with-jpeg --with-tiff --with-png --with-gtk
make bootstrap
make install
```

Опция **--with-gtk** говорит о том, что *Emacs* будет собран с поддержкой GTK2. Опции **--with-xpm --with-jpeg --with-tiff** и **--with-png**, в принципе, можно и не указывать – если вам не нужна поддержка соответствующих графических форматов (да, в *Emacs* можно смотреть картинки!).



***Emacs* по умолчанию.**

```
(custom-set-variables
  ;; custom-set-variables was added by Custom -- don't edit or cut/paste it!
  ;; Your init file should contain only one such instance.
 '(case-fold-search t)
 '(completion-on-separator-character t)
 '(current-language-environment "Cyrillic-KOI8")
 '(default-input-method "cyrillic-jcuken")
 '(ecb-change-layout-preserves-compwin-state t)
 '(ecb-compile-window-height 8)
 '(ecb-compile-window-prevent-shrink-below-height t)
 '(ecb-compile-window-width (quote edit-window))
 '(ecb-layout-name "left3")
 '(ecb-options-version "2.32")
 '(ecb-primary-secondary-mouse-buttons (quote mouse-1--mouse-2))
 '(ecb-source-path (quote ("/home/httpd/html" "/data/prj" "/")))
 '(ecb-tip-of-the-day nil)
 '(ecb-tree-expand-symbol-before t)
 '(ecb-windows-width 0.25)
 '(global-font-lock-mode t nil (font-lock))
 '(php-manual-path "/data/docs/php/manual")
 '(php-manual-url "/data/docs/php/manual/")
 '(show-paren-mode t nil (paren))
 '(transient-mark-mode t))
(custom-set-faces
  ;; custom-set-faces was added by Custom -- don't edit or cut/paste it!
  ;; Your init file should contain only one such instance.
 '(show-paren-mismatch-face ((((class color)) (:background "red" :foreground "white")))))
--:--  .emacs          (Emacs-Lisp)--L1--Top-----------------------------------
```

***Emacs* минималиста.**

минибуфером, появится приглашение к вводу имени файла для редактирования. Вы можете ввести имя как существующего файла (тогда он откроется для редактирования), так и не существующего. В последнем случае *Emacs* откроет пустой буфер, который впоследствии будет сохранен в указанный файл.

Буфер – одно из ключевых понятий *Emacs*, он представляет собой просто текст, открытый в данный момент в *Emacs*. На самом деле, это определение довольно неточное, но для начала сойдет: проникнувшись философией *Emacs*, вы и сами увидите, что буфер может содержать что угодно – и таблицу процессов, и текст, и даже PNG-картинку. По умолчанию при открытии *Emacs* создает два буфера – scratch и Messages. О первом вы можете прочесть в соответствующей врезке, а второй содержит отладочные сообщения редактора.

Минибуфер поддерживает функцию автодополнения имен файлов/каталогов по клавише Tab, на манер командной оболочки Bash.

Думаю, с открытием файла вы разобрались. Теперь попробуйте открыть другой файл, или создать новый. Как видите, он открылся, но куда делся предыдущий? Никуда – он просто находится в другом буфере! При открытии или создании файла *Emacs* создает для него новый буфер, который становится активным. Для переключения между буферами используется комбинация **C-x C-b**, после нажатия которой в минибуфере появится приглашение к вводу имени буфера, на который вы хотите переключится. Если вместо ввода имени буфера просто нажать Enter, то *Emacs* переместит вас к предыдущему буферу.

Для выхода из редактора служит комбинация **C-x C-c**. Если на момент выхода у вас будут измененные буферы, которые вы не сохранили, обратите внимания на строку минибуфера – *Emacs* обязательно спросит, не хотите ли вы сохранить изменения. Для закрытия одного лишь активного буфера применяется сочетание клавиш **C-x C-k**.

Окно редактирования *Emacs* можно заставить отображать сразу несколько буферов, разделив его на несколько независимых частей. Для разделения окна по горизонтали нажмите **C-x 2**, по вертикали – **C-x 3**. Перемещаться между различными частями окна редактирования можно при помощи сочетания **C-x o**. Для того, чтобы вернуться в обычный режим с отображением одного буфера в окне редактирования, нажмите **C-x 1**. Обратите внимание, что активным станет тот буфер, в которым вы находились до нажатия **C-x 1**.

Конечно же, не все команды *Emacs* имеют горячие клавиши – функций у редактора слишком много, да и каждое установленное дополнение добавляет свои собственные. Поэтому все команды можно вводить в минибуфере вручную. Если вы просто щелкните мышью на строке минибуфера, ничего путного из этого не выйдет. Для выполнения внутренних команд есть комбинация **M-x** (напомню, это Alt-x или последовательное нажатие Esc и x). Давайте потренируемся. Нажмите **M-x**, введите в минибуфер команду save-buffer (для имен команд также работает функция автодополнения) и нажмите Enter. В результате активный буфер (то есть редактируемый в данный момент) будет сохранен в файл. Как вы уже, наверное, догадались, именно эта команда привязана к комбинации **C-x C-s**.

Если вы случайно нажмете **C-x** или, например **M-x**, или просто передумаете вызывать какую-нибудь внутреннюю команду, нажмите **C-g**. После этого информация в минибуфере уничтожится, а сам он станет неактивным.

Процесс копирования-вставки в *Emacs* имеет свои особенности. Для копирования куска текста установите курсор на его начало и нажмите **M**-пробел. В минибуфере в это время должно появиться сообщение Mark Set («метка установлена»). Теперь с помощью клавиш-стрелок, PgUp/PgDown или Home/End доберитесь до конца нужного куска и нажмите **C-w**. Это команда вырезает часть текста и заносит его в буфер обмена. Для копирования нажмите вместо **C-w M-w**, а для вставки из буфера обмена – **C-y**. Конечно, можно вместо C-пробел использовать для выделения мышь, но быть может вы такой же мышефоб, как и я? Когда вы выделяете текст мышью, он подсвечивается, а при выделении с помощью комбинации **C**-пробел этого, по умолчанию, не происходит. Это можно исправить, активировав в главном меню опцию Active Region Highlighting (Transient Mark Mode), которая находится в подменю Options.

Путь к СУТРМ

Ну что ж, думаю, самые основы работы с *Emacs* вы постигли, давайте же превратим его в СУТРМ, то есть в Самый Удобный Текстовый Редактор в Мире. Во-первых, удалите файл *.emacs* из вашего домашнего каталога, если он там есть и перезапустите редактор – начнем, как говорится, с чистого листа.

Во-первых стоит рассказать *Emacs'у*, что мы по умолчанию хотим сохранять наши файлы в кодировке koi8-r. Для этого выберите в главном меню Options->Mule (Multilingual Environment)->Set Language Environment->Cyrillic->Cyrillic-KOI8. Теперь, если хотите, включите подсветку активного участка текста: Options->Active Region Highlighting (Transient Mark Mode) и подсветку парных скобок: (Options->Paren Match Highlighting (Show Paren Mode). Это, пожалуй, все, что лично я доверяю «автоматике» *Emacs*'а. Сохраните



БУФЕР *SCRATCH*

Многие специально отключают этот буфер, а зря: никаких неудобств он не создает, а вот кое-какую пользу из него извлечь можно. Содержащийся в нем текст гласит, что его можно использовать для временного хранения какого-нибудь текста или для выполнения Lisp-операций. То есть это простейший интерфейс для *Emacs* Lisp. Попробуйте, например, ввести такое выражение:

```
(+ 3 4)
```

и для перехода на новую строку нажать не Enter, а **C-j**. Как видите, интерпретатор Elisp выдал ответ: 7. Таким образом, этот буфер можно использовать как продвинутый калькулятор с поддержкой переменных, сложных функций, рекурсии. Если, конечно, привыкните к обратной польской записи и скобкам, принятым во всех диалектах Lisp. Попробуйте еще несколько примеров:

```
(- (+ 5 9) (- 4 9))
(* (+ (/ 90 10) (+ 2 3)) (sqrt 4))
(setq g 9.8)
(* 5 g)
```

НАЧИНАЮЩЕМУ ЕМАКСЕРУ

Действие	Клавиатурная комбинация	Название функции
Создать/Открыть файл	C-x C-f	find-file
Сохранить буфер в файл	C-x C-s	save-buffer
Сохранить буфер под новым именем	C-x C-w	save-with-newname
Сохранить все измененные буферы в файлы	C-x s	save-some-buffers
Переключиться в другой буфер	C-x C-b	switch-to-buffer
Разделить окно редактирования и открыть другой буфер	—	switch-to-buffer-other-window
Закрыть активный буфер	C-x k	kill-buffer
Установить метку	C-<пробел>	set-mark-command
Вырезать выделенный текст	C-w	kill-region
Скопировать выделенный текст	M-w	copy-region-as-kill
Вставить скопированый/вырезаный текст	C-y	yank
Разделить окно по горизонтали	C-x 3	split-window-horizontally
Разделить окно по вертикали	C-x 2	split-window-vertically
Поиск текста с подсветкой всех найденных участков	C-s	isearch-forward
Отменить последнюю операцию	C-x u	undo
Выйти из минибуфера	C-g	keyboard-quit
Закрыть Emacs	C-x C-c	save-buffers-kill-emacs



эти настройки, выбрав в меню Options->Save Options, дальше мы будем редактировать конфигурационный файл вручную.

Главным конфигурационным файлом *Emacs*, как вы уже поняли, является .emacs, находящийся в домашнем каталоге. Он, кстати, представляет собой самый настоящий скрипт на языке *Emacs Lisp* – именно поэтому его лучше редактировать вручную, тем более что в главном меню *Emacs* доступно меньше 10% всех настроек *Emacs*.

Не кажется ли вам панель инструментов *Emacs* бесполезной? Мне вот, например, кажется, тем более, что клавиатурные сочетания всех кнопок, находящихся на ней, можно запомнить меньше чем за 10 минут. Для того, чтобы панель не появлялась при запуске *Emacs*, в **.emacs** нужно добавить такую строку:

```
(tool-bar-mode nil)
```

Истинные минималисты вроде меня могут добавить еще **(menu-bar-mode nil)** для скрытия главного меню и **(scroll-bar-mode nil)** для удаления полос прокрутки. К меню всегда можно обратиться с помощью щелчка правой кнопки мыши в буфере при зажатом Ctrl. Кстати, переключаться между буферами можно таким же образом, только вместо правой кнопки используйте левую.

Я знаю, вас раздражают эти **C-x C-f** и **C-x C-s**. Давайте же «перевесим» команды открытия и сохранения файла на другие комбинации. Добавьте следующие строки в свой **.emacs**:

```
(global-set-key «\C-s» 'save-buffer)
(global-set-key «\C-o» 'find-file)
(global-set-key «\M-f» 'isearch-forward)
```

Таким образом мы назначили команды сохранения (save-buffer) и открытия файла (find-file) на **C-s** и **C-o** соответственно. Но зачем мы назначили на **M-f** какой-то isearch-forward? На самом деле это очень удобная функция линейного поиска по тексту, которая по умолчанию привязана к **C-s** (таким образом, без этой строки мы могли бы ее использовать лишь набрав вручную имя в минибуфере). Для поклонников mcedit можно порекомендовать такую конфигурацию:

```
(global-set-key [f2] 'save-buffer)
(global-set-key [f1] 'find-file)
(global-set-key [f7] 'isearch-forward)
```

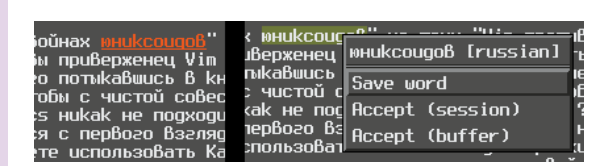
Подобным образом можно назначить любую комбинацию для любой функции. Главное, не забывайте обрамлять сочетания клавиш двойными кавычками (к функциональным клавишам это не относится) и ставить перед именами функций апострофы.

Наводим марафет

Вряд ли вас устраивает цветовая схема *Emacs*, которая установлена по умолчанию, да и шрифты никуда не годятся: латинские одного размера, кириллические – другого. Нужно исправлять ситуацию. Для изменения цветов редактора предусмотрены функции **set-face-background** и **set-face-foreground**. Первая отвечает за фон, вторая – за цвет расположенных на нем объектов. Например, в **.emacs** следующих строк:

```
(set-face-background 'default «red»)
(set-face-foreground 'default «blue»)
```

заставит *Emacs* выводить текст красными буквами на синем фоне (адское сочетание! Не повторяйте этого дома, ну или хотя бы предварительно к окулисту запишитесь). Вместо «red» или «blue» можно использовать цвета вроде «#3743ec», так что откройте палитру в *Gimp* и не ограничивайте себя всякими «green» и «black». Для изменения цветов меню нужно заменить **'default** на **'menu**.



Ispell не знает слова «юниксоид». Непростительная ошибка!

N ПРИЧИН, ПО КОТОРЫМ EMACS ЛУЧШЕ VIM

Думаю, после прочтения статьи приверженцы редактора *Vim* очень хотят услышать ответ на вопрос, чем же все-таки *Emacs* лучше, чем *Vim*.

Около полугода назад у меня были установлены оба этих редактора, и оба казались мне жутко неудобными. Для правки конфигурационных файлов я предпочитал использовать nano, а для редактирования исходных текстов – *mcedit*. Так и продолжалось, пока я не стал изучать документацию по *Vim* и *Emacs*. Обнаружилась интересная вещь – к *Vim* в его первоначальном виде привыкнуть гораздо легче, да и к сносному редактированию чего угодно он готов сразу же. *Emacs* же требует настройки любой мелочи, зато его можно превратить хоть в nano, хоть в IDE Visual C++.

По количеству расширений *Emacs* оставляет *Vim* далеко позади. Да и функциональность *Vim'овских* скриптов по сравнению с расширениями *Emacs* оставляет желать лучшего. Так, например, *Vim'овского* аналога *Emacs Code Browser* (ecb) и LaTeX-preview мне найти так и не удалось, как не удалось найти и других, не столь значимых для меня вещей. *Emacs* лучше интегрируется со сторонними приложениями: попробуйте заставить *Vim* работать с *Maxima* или *Gnuplot*. В основном это заслуга *Emacs Lisp*, на котором очень просто дописываются те функции, которых вам не хватает.

Если редактор вам нужен только для правки конфигурационных файлов, то вам одинаково подойдут и *Emacs*, и *Vim*. Если же вы намерены получить от текстового редактора нечто большее, то просто установите и тот, и другой, и вы сами без труда найдете эти самые N причин. Ну а если вы не хотите тратить время на чтение документации и настройку «какого-то там текстового редактора», то может быть стоит попробовать *jEdit*?

IDO.EL

Несмотря на функцию автодополнения, вас наверняка раздражает отсутствие в *Emacs* обычного диалога открытия файла. Но ведь будь в *Emacs* такой диалог, любители делать-все-только-с-клавиатуры справедливо бы возмутились. Тем не менее, к *Emacs* есть отличное дополнение, делающее открытие файла и переключение между буферами невероятно простым и приятным. Называется оно ido, и, как и все дополнения к *Emacs*, представляет собой обычный Elisp-скрипт. Получить **ido.el** можно на сайте разработчика **http://www.cua.dk**. Давайте рассмотрим установку этого дополнения.

Для активации и использования функций любого дополнения *Emacs*, его нужно поместить в директорию автоматической загрузки (load-path). Рекомендую вам создать в своем домашнем каталоге директорию emacs, в которой будут располагаться дополнения. Скопируйте **ido.el** в **~/emacs/**, а в .emacs добавьте следующие строки:

```
(add-to-list 'load-path «~/emacs»)
(require 'ido)
```

Первая строка добавляет **~/emacs** в список директорий автозагрузки, вторая говорит о том, что при запуске редактора необходимо загрузить файл **ido.el**. Это значит, что теперь в редакторе стали доступны функции дополнения **ido.el** – в данном случае их имена начинаются с **ido-**. Однако для активации дополнения этого недостаточно, нужно еще включить **ido**-режим:

```
(ido-mode t)
```

Слово «t» в различных диалектах языка Lisp означает «true». Тем самым вы включили дополнение, и теперь можете посмотреть, как будет огранизовано открытие файла (нажмите **C-x C-f**). Отныне к этой комбинации привязана функция **ido-find-file** (это произошло после включение режима **ido-mode**). Для перемещения между директориями в **ido-find-file** можно использовать кнопки-стрелки, для выбора файла/директории – Enter, для выхода в родительский каталог – Backspace. Намного удобнее, чем обычный **find-file**, не правда ли? Подобным же образом теперь работает и переключение между буферами (**C-x C-b**).

Если вы будете постоянно пользоваться этим дополнением, логично было бы переназначить функции **ido-find-file** и **ido-switch-buffer** на более удобные комбинации:

```
(global-set-key [f1] 'ido-find-file)
(global-set-key [f11] 'ido-switch-buffer)
```

```
-R:--  *scratch*         (Lisp Interaction)--L5--All------------------
 Find file: /data/{articles/ | arch/ | big/ | docs/ | drv/ | games/ |
 html/ | music/ | obrezh.ru/ | pics/ | prj/ | univer/ | ...}
--:--  WindowMaker        (Fundamental)--L1--Top-----------------------
 Buffer: {.emacs | *scratch* | *Messages* | WindowMaker}
```

Вот так преобразился процесс открытия файлов и переключения между буферами.

Я предпочитаю использовать немного другой способ, который потребует значительно меньше телодвижений. Наверняка в вашем домашнем каталоге есть файл с описанием параметров ресурсов *X*-Window (он обычно называется *.Xresources* или **.Xdefaults**). Если его нет – создайте его и внесите в этот файл всего две строки:

```
Emacs*background: #3d3d3d
Emacs*foreground: #e1e1e1
```

Теперь запустите **xrdb ~/.Xresources** для обновления параметров ресурсов. После этого и в меню, и в панели инструментов, и в буфере фоном будет «#3d3d3d», а цветом переднего плана (шрифта) – «#e1e1e1». Гораздо удобнее, не правда ли?

Так же дело обстоит и со шрифтами: есть функции **set-face-font**, работающие таким же образом, как и **set-face-background**, но удобнее использовать **.Xresources**. Мне нравится полужирный шрифт terminus:

```
Emacs*font: -xos4-terminus-bold-r-normal--14-140-72-72-c-80-koi8-r
```

Для выбора шрифта (а заодно и корректного составления его имени) рекомендую воспользоваться утилитой *xfontsel*, входящей в комплект *X*-Window.

Что ж, надеюсь, теперь вам приятнее смотреть на ваш *Emacs*...

Проверка орфографии

Что еще не помешало бы текстовому редактору? Наверное, проверка орфографии. В *Emacs* для проверки орфографии используется внешняя утилита *ispell*. Плюс ко всему существует специальный режим – **flyspell-mode** (для его работы также необходима *ispell*), который сделан на манер проверки орфографии в текстовых процессорах: вы печатаете слово, а *flyspell* сразу же проверяет его. Если такое слово не найдено в словаре, оно выделяется и подчеркивается. Щелкнув на подчеркнутом слове средней кнопкой мыши, вы сможете выбрать нужный вариант из тех, что подобрал *flyspell* или занести это слово в словарь. Разумеется, для работы *flyspell* вы должны установить в своей системе пакеты *ispell* и *ispell-ru*. Если *ispell* есть практически в любом дистрибутиве, то с *ispell-ru* (который содержит русский словарь для *ispell*) не все так гладко. Если вы не найдете его в своем дистрибутиве, вам придется скачать один из словарей отсюда: **http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html**.

После этого нужно указать *Emacs*, чтобы он использовал для проверки орфографии русский словарь. За имя словаря, используемого *ispell* и *flyspell*, отвечает внутренняя переменная **ispell-dictionary**. Очевидно, необходимо установить для нее значение «russian». Есть несколько способов сделать это. Если хотите установить какую-либо переменную только для текущего сеанса, то для этого прекрасно подойдет функция **customize-variable**. Но если вы захотите, например, всегда использовать русский словарь, то проще прописать значение соответствующей переменной в файл **.emacs**:

```
(setq flyspell-dictionary «russian»)
```

Давайте назначим командам проверки орфографии какие-нибудь простые комбинации:

```
(global-set-key [f11] 'ispell-buffer)
(global-set-key [f12] 'flyspell-mode)
```

Теперь по клавише F12 вы сможете включать/выключать режим проверки «на лету», а по F11 производить пошаговую проверку.

Возможно у читателя появится вопрос – а зачем нужно возиться с *Emacs*, если тот же *Vim*/*Gedit*/*Kate* подходит для удобного редактирования текста практически «из коробки»? Да, но ведь к *Vim'y* тоже нужно привыкать, а без документации использование *Vim* сродни ловле черного кота в темной комнате – можно ориентироваться только по звуку. Ну и наконец, если вы являетесь приверженцем *Vim*, почему бы просто из любопытства не посмотреть на то, как ваши любимые функции реализованы в *Emacs*? *Kate*/*Gedit* конечно же очень просты, однако в них нет и двадцатой части функций *Emacs*.

Да, *Emacs* довольно непрост в конфигурировании, но зато настройке поддается абсолютно все!

К сожалению, современной русскоязычной документации по *Emacs* не существует, есть лишь перевод руководства к уже давно устаревшей 19-й версии от Алекса Отта (см. **http://linux.org.ru/books/emacs-manual**). Впрочем, разобраться с *Emacs* можно и без официального руководства, благо теперь вы знаете основы, так что дальше дело пойдет быстрее. Некоторое количество полезной информации можно почерпнуть из *Emacs wiki* (**http://www.emacswiki.org**), на которой есть и русскоязычный раздел.

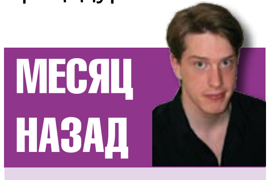
В крайнем случае, если вам что-то непонятно, пишите мне на **rc_4096@list.ru**. LXF

MYSQL 5.0 SPECIAL SCRIPT PROGRAMMING

PHP: Хранимые процедуры



ЧАСТЬ 2 Как будто нам недостаточно изучения SQL! *MySQL 5.0* приносит с собой ещё и новый способ программирования. Специалист по базам данным **Пол Хадсон** (Paul Hudson) представляет хранимые процедуры.



Мы познакомились с транзакциями и триггерами

Я хотел бы сделать важное заявление: PHP не всегда самый лучший язык программирования. Да, как правило, он лучший, но кроме него существует ещё и множество других, подходящих в каждой конкретной ситуации. А при использовании нескольких языков одновременно сразу же возникает вопрос – как правильно взаимодействовать с базой данных. Возможно, вы мне не поверите, но с *MySQL 5.0* такой проблемы не существует.

Условия задачи таковы: не важно, какой язык программирования вы используете, информация в базе данных должна быть одна и та же. Если вы работаете с базой данных вашей коллекции компакт-дисков, скорее всего, вам не придётся особенно заботиться о методах доступа к информации, достаточно просто прочитать несколько строк и, возможно, сделать некоторые изменения. Но если вы разрабатываете приложения для работы с банковской базой данных, то просто необходимо всегда следовать строгим правилам, выполнять все необходимые проверки, вести исчерпывающий журнал всех действий и так далее. Очень сложно контролировать соблюдение всех этих правил в среде, оперирующей различными языками программирования.

Теперь у нас появилось решение! Вам больше не нужно думать о том, чтобы помногу раз копировать тысячи строк SQL-кода. Поддержка множества копий одного и того же запроса – это жуткий кошмар, тысячи человеко-часов и питательная среда для ошибок. Нет, самым лучшим вариантом было бы сохранить SQL прямо в базе данных и в дальнейшем вызывать его изо всех приложений. Вместо того, чтобы поддерживать код, который загружает информацию о пользователе 50235, выполняет нужные изменения и сохраняет запись о них в журнале, гораздо лучше записать всю последовательность действий в базу данных в виде функции и затем вызывать её примерно так: **updateuser(50235)**. При использовании этого метода код на любом языке программирования вообще не должен знать, что требуется сохранять что-то в журнале, не говоря уж о том, что именно надо туда записывать: база данных сама позаботится об этом. Если же правила затем изменятся, вам потребуется внести изменения в базу данных и никуда больше, вместо того чтобы править сотни файлов на разных языках программирования.

Это решение известно как хранимые процедуры. Если вы найдёте время на то, чтобы выучить их синтаксис, то получите ценное приобретение в ваш арсенал администратора базы данных. В предыдущем номере мы с вами рассматривали, как создавать и использовать триггеры на таблицах, так что теперь вы умеете наблюдать за выполняемыми запросами и контролировать их. Хорошая новость: триггеры – это не более чем хранимые процедуры, разработанные для запуска в специфические моменты. Таким образом, триггеры – это подмножество хранимых процедур, не позволяющее получать и передавать ваши переменные. Так что если вы изучили учебник из февральского номера, у вас уже есть фора.

ГДЕ ТУТ ХРАНИМЫЕ ПРОЦЕДУРЫ?

Новые возможности *MySQL 5.0* позволяют создавать хранимые процедуры, используя диалект SQL под названием SQL:2003. Сейчас идёт работа по созданию системы подключаемых хранимых процедур, которая представляет собой открытый API, позволяющий разрабатывать хранимые процедуры на любом языке программирования.

Разработчики *MySQL* утверждают, что, скорее всего одним из первых поддерживаемых языков программирования будет PHP (Урааа!), поскольку он разработан так, что его довольно просто внедрять. Если они справятся с этой задачей, мы сможем из сценариев на PHP вызывать хранимые процедуры *MySQL*, которые в свою очередь написаны на PHP. Они смогут выполнять обратные вызовы к базе данных и выполнять серьёзную обработку перед тем, как вернуть окончательный результат. Если вас напрягает, что какой-то язык X не умеет делать что-то, что умеет PHP, то такие хранимые процедуры могут стать решением вашей проблемы...

Следующие процедуры

Давайте рассмотрим простую хранимую процедуру, которая умеет обновлять возраст пользователя. Для начала создадим таблицу пользователей, примерно вот так:

```
CREATE TABLE users (ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, Name VARCHAR(255), Age TINYINT UNSIGNED);
INSERT INTO users (Name, Age) VALUES ('Nick', 44);
```

Процедура будет менять возраст пользователя с заданным именем, делая его на год старше. Как и с триггерами, нам потребуется изменить разделитель команд с ; на // для того, чтобы ввести многострочную процедуру как единое целое.

Итак:

```
DELIMITER //
CREATE PROCEDURE get_older (IN pName VARCHAR(255))
BEGIN
UPDATE users SET Age = Age + 1 WHERE Name = pName;
END//
DELIMITER ;
```

После смены разделителя мы создаём процедуру **get_older** и объявляем, что она принимает параметр типа **VARCHAR(255)** под названием **pName**. Обратите внимание, параметры **BEGIN** и **END** используются для определения границ процедуры, но их можно опускать, если процедура состоит из одной единственной строки (как у нас). Наша процедура обновляет все строки в таблице, удовлетворяющие указанному условию (зависящему от значения параметра), увеличивая значение поля **age** на единицу. В конце определения процедуры мы возвращаем определение разделителя к стандартному значению.

Как вызвать нашу процедуру? Нет ничего проще.

```
SELECT * FROM users;
CALL get_older('Nick');
SELECT * FROM users;
```

Два оператора **SELECT** нужны для того, чтобы увидеть, как изменилось значение **Age** после вызова **get_older()**, для работы хранимой

процедуры они не нужны. Как видите, **get_older()** работает именно так, как вы ожидали, и её достаточно просто использовать. Если вас интересует, что будет, если передать параметр, который не является **VARCHAR(255)**, то ответ прост – *MySQL* преобразует его к типу **VARCHAR**.

Чтение данных

Переместимся в более сложные области и добавим ещё одного пользователя в нашу таблицу:

```
INSERT INTO users (Name, Age) VALUES ('Nick', 32);
SELECT * FROM users;
```

Теперь у нас есть два Ника, с ID равным 1 и 2. Мы можем создать хранимую процедуру для возвращения значений из запроса **SELECT** очень простым способом.

```
DELIMITER //
CREATE PROCEDURE get_user (IN pID VARCHAR(255))
BEGIN
SELECT * FROM users WHERE ID = pID;
END//
DELIMITER ;
```

Здесь используется та же самая идея, за тем исключением, что мы указываем ID для точного определения пользователя и используем оператор *SELECT*, чтобы извлечь информацию из таблицы. Вызов **get_user()** в данном случае возвратит одну строку, но если бы так случилось, что к условию подошли бы несколько строк, то все они были бы получены.

Второй способ вернуть данные из процедуры – это использование выходных (**OUT**) параметров. До сих пор мы помечали все параметры как **IN**, что на самом деле не обязательно, поскольку *MySQL* считает все параметры входящими по умолчанию. Если вы когда-нибудь использовали CORBA, COM+ или D-BUS, то понятие **OUT**-параметров вам знакомо. Если же нет, то всё станет очевидно после следующего примера. Давайте попробуем вернуть значение из процедуры при помощи выходного параметра:

```
DELIMITER //
CREATE PROCEDURE get_user (IN pName VARCHAR(255),
OUT pID INT)
BEGIN
SELECT ID FROM users WHERE Name = pName LIMIT 1
INTO pID;
END//
DELIMITER ;
```

Важным является тот момент, что мы немного изменили работу запроса, добавив оператор **LIMIT 1**. Он потребовался по той причине, что в качестве **OUT**-параметра мы можем вернуть одно и только одно значение, и если SQL-запрос вдруг вернёт несколько строк, то произойдёт ошибка. Вызов этой процедуры немного отличается от приведённого выше, так как нам надо передать два параметра, один из которых является переменной SQL и может быть опрошен позднее. Итак:

```
mysql> CALL get_user('Nick', @ID);
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT @ID;
+------+
| @ID |
+------+
| 1 |
+------+
1 row in set (0.00 sec)
```

Inline-функции

Процедуры хорошо работают, если вам нужно изменить несколько вещей одной строкой кода, но из них очень неудобно возвращать значения переменных.

Как Pascal или Fortran, *MySQL* различает процедуры и функции. Первые – это команды, а вторые – запросы. Если вы изучали компьютерные науки, то вам должно быть знакомо понятие «Command-Query Separation".

В *MySQL* процедуры не могут быть вызваны inline, и хотя они умеют возвращать значения при помощи **OUT**-переменных, вам потребуется выполнить ещё один запрос, чтобы прочитать их. Альтернативный вариант - использовать функции, разработанные так, что их можно вызывать как часть более длинного SQL-запроса, наподобие встроенных функций **MIN()** и **MAX()**. Чтобы протестировать, как это работает, давайте добавим две следующие строчки в таблицу users:

```
INSERT INTO users (Name, Age) VALUES ('Tanya', 45);
INSERT INTO users (Name, Age) VALUES ('William', 39);
```

Итак, у нас теперь есть Nick и Tanya старше 40 лет, плюс William и ещё один Nick моложе сорока. Мы могли бы отфильтровать его по условию **"WHERE Age > 40"**, но в демонстрационных целях сделаем это при помощи хранимой процедуры **age_over_40()**. Вот как это будет выглядеть:

```
DELIMITER //
CREATE FUNCTION age_over_40 (pAge INT) RETURNS
BOOLEAN
BEGIN
IF pAge > 40 THEN
RETURN TRUE;
ELSE
RETURN FALSE;
END IF;
END//
DELIMITER ;
```

Этот пример немного отличается от приведённого выше, так что давайте рассмотрим, в чём разница. Во-первых, теперь нет никаких **IN** или **OUT** – все параметры функции являются входными. Но зато теперь мы должны определить возвращаемое значение при помощи оператора **RETURN**. В нашем примере мы возвращаем логическую (Boolean) переменную, которая может принимать значения True или False. Внутри функции мы с вами впервые использовали условный оператор, и проверили, превышает ли значение переданного нам параметра 40. Если да, то мы возвращаем true, если нет – то false. Просто!

Самая потрясающая вещь в функциях – это то, что их можно использовать как часть обычных SQL-запросов, например, вот так:

```
SELECT Name FROM users WHERE age_over_40(Age);
```

Этот запрос вернёт все значения **Name** из таблицы **Users**, для которых функция **age_over_40()** вернула значение true. Представьте всё богатство возможностей – вы можете запросить несколько таблиц, написать сложные вложенные условия с множеством проверок, и так далее... Если вам нужны более сложные проверки, возможно, вам больше подойдёт оператор **CASE**:

```
DELIMITER //
CREATE FUNCTION describe_name (pName VARCHAR(255))
RETURNS VARCHAR(255)
BEGIN
CASE pName
WHEN 'Nick' THEN RETURN 'Short for Nicholas';
WHEN 'William' THEN RETURN 'Often shortened to Will or
Bill';
ELSE RETURN 'I've never heard of that name!';
END CASE;
END//
DELIMITER ;
```

Для самых продвинутых пользователей существуют циклы, но вам стоит хорошенько подумать, чтобы найти для них достойное применение.

Вот и всё, друзья, что я хотел сказать на тему хранимых процедур (и хранимых функций!). Надеюсь, вы увидели, как полезно может быть инкапсуляция сложных запросов в однострочное выражение. LXF

ПОДСКАЗКИ

- Если вам нужно вернуть несколько значений сразу, то вы можете использовать столько OUT-параметров, сколько хотите. Для этого просто напишите оператор SELECT ... INTO и укажите в нём список из всех переменных, например "INTO foo, bar".
- Создавать функции с именами функций, уже существующих в *MySQL* – это не очень хорошая идея. Кроме того, у вас возникнут проблемы, если имена переменных совпадут с названиями таблиц или отдельных полей в них.
- Система преобразования типов *MySQL* может успешно преобразовать друг в друга переменные почти всех типов данных, так что если ваша хранимая процедура ожидает параметр типа **VARCHAR(255)**, а вы передадите ей **CHAR(50)**, то никаких вопросов не возникнет. Но не забывайте, что любое преобразование типов сказывается на производительности, поэтому почему бы всегда не передавать процедурам именно то, что вы и планировали им передавать с самого начала?



ЧЕРЕЗ МЕСЯЦ

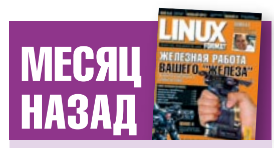
Мы закончим наше исследование *MySQL 5*, изучив ограниченный доступ к базе данных через представления.

HARDCORE LINUX СЕРИЯ «LINUX ДЛЯ СПЕЦИАЛИСТОВ»

Autopackage Создаем пакет

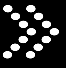
Если вы сумеете постичь сложный процесс создания файлов *Autopackage*, линуксоиды всего мира будут вам благодарны. **Грэм Моррисон** покажет, как это делается.


НА ДИСКЕ



МЕСЯЦ НАЗАД

Мы узнали, как помочь сообществу, написав man-страницу для любимого приложения



Linux остро нуждается в более простом способе установки программ. Новички часто бывают обескуражены, когда вместо установки программы щелчком мыши по **setup.exe** им приходится брести по тернистой дистрибутиво-зависимой тропе установки пакетов, на ходу обучаясь работе с пакетными менеджерами, RPM и DEB-файлами и разбираясь с адской кухней зависимостей. Безусловно, это одна из главных причин, почему начинающие пользователи теряют мужество и возвращаются в Windows. Да и Linux-ветеранов раздражает, что очередную новинку нельзя по-быстрому установить и поглядеть без отслеживания дюжины зависимостей.

Текущая ситуация – это неустойчивое равновесие. В каждом дистрибутиве поддерживаются свои пакеты, которые исправляются и обновляются. Пользователи вынуждены переходить на новые версии дистрибутивов, если им нужны новые пакеты, либо искать альтернативы. Чаще всего такой альтернативой является самостоятельная сборка пакета из исходных текстов, но не у всех пользователей есть на это время.

Autopackage пытается внести порядок в этот хаос, посулив предоставить Linux альтернативу **setup.exe**, то есть пре-компилированные бинарные пакеты и автоматическое разрешение зависимостей. *Autopackage* ориентирован на совместимость с максимально большим числом дистрибутивов, причем установка везде проходит одинаково. Если вы предусмотрели *Autopackage*-файл для программы, вы можете быть уверены, что люди смогут установить ее легким движением руки независимо от используемого дистрибутива.

Работа для умелых рук…

Расплатой за легкость установки программы из *Autopackage* является довольно сложный процесс создания такого пакета. Пусть не вы писали программу, но, создав файл *Autopackage*, вы можете существенно облегчить жизнь ее пользователям и разработчику. Здесь не обойтись без хорошего знания Linux, включая версии установленных у вас библиотек и зависимостей выбранной вами программы. Да еще неплохо бы уметь программировать, поскольку исходный код программы скорее всего придется подправить для уменьшения числа зависимостей, замены опций компилятора или улучшения совместимости с различными версиями библиотек. Другого пути нет, в любом случае вы должны уверенно чувствовать себя в Linux. Фактически, *Autopackage* позволяет Linux-хакерам совершенствовать мастерство.

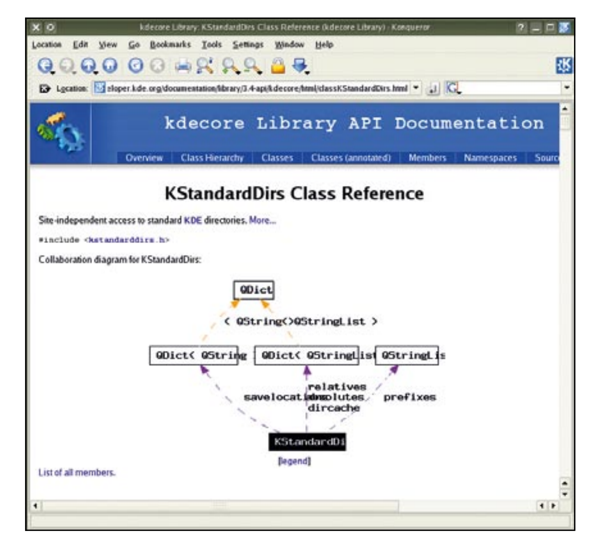
В процессе урока я шаг за шагом покажу вам на реальном примере, как создавать *Autopackage*-файл (далее Автопакет). Для этой цели я выбрал программу Kalbum, каталогизатор фотографий – слегка облегчив вам задачу, поскольку у *Kalbum* сравнительно мало зависимостей. Чем больше зависимостей, тем сложнее задача, и тем больше знаний требуется для ее решения.

Разобьем процесс создания автопакета на несколько шагов. Во-первых, нужно убедиться, что приложение в состоянии запускаться из любого каталога на жестком диске, иначе с ним не смогут работать те, у кого нет прав суперпользователя. Затем, с помощью опций сборки мы сократим зависимости, насколько возможно, после чего создадим spec-файл для завершения построения автопакета.

Перед тем, как начать, установите утилиты разработки *Autopackage* (**http://autopackage.org/download-tools.html**).

В данном уроке я использую версию *Kalbum 1.0*, ее можно взять с диска к *LXF72* или с **www.paldandy.com/kalbum**.

ШАГ 1 – СДЕЛАТЬ ПРИЛОЖЕНИЕ ПЕРЕМЕЩАЕМЫМ



В API KDE предусмотрен стандартный класс для поиска файла.

Для максимальной гибкости нужно сделать приложение перемещаемым. То есть программа обязана запускаться в любом случае, независимо от того, где она установлена. Тогда пользователи, не имеющие root-доступа к рабочей машине, смогут установить ее в свою домашнюю директорию. Но как узнать, насколько перемещаемо приложение?

Только путем чтения исходного кода. Как раз в нем разработчики указывают, требует ли программа прав суперпользователя (кстати, имейте это в виду, если затеете писать собственное приложение). Разработчики *Autopackage* специально создали утилиту BinReloc, чтобы хоть немного упростить мучительный процесс перемещения программы. Утилита статически компонуется с вашим приложением и зависит только от *libc*.

Преимущество KDE-приложений вроде *Kalbum* – наличие класса KStandardDirs, он выполняет ту же функцию, что и BinReloc: доступ к стандартным системным каталогам изнутри вашей программы. Так что править исходный код в данном случае нам не придется, и можно переходить к следующему шагу. (Отметим, что разработчики *Autopackage* надеются в будущем сделать сборку перемещаемых пакетов почти автоматической). Давайте посмотрим, как класс KStandardDirs используется в *Kalbum* (файл **photodetails.cpp**):

```
QString dir;
KGlobal::dirs()->addResourceType(«kalbum_data», KGlobal::dirs()->kde_default(«data») + «kalbum/data/»);
dir = KGlobal::dirs()->findResourceDir(«kalbum_data», «slides/slides.png»);
```

Этот кусок кода использует метод addResourceType класса KStandardDirs (типа 'данные', в отличие от, например, иконок), определяет директорию, а директория затем присваивается строке (dir), вместе с вложенной директорией и именем файла, который мы будем в ней искать. Строка укажет **slides.png**, расположена программа в домашней директории пользователя или в системном каталоге, тем самым выполняя функцию BinReloc. API *Autopackage* также включает собственные команды для определения местоположения программ в файловой системе и создания директорий, чтоб не пришлось разбираться, локальное имя файла или глобальное.

ШАГ 2 – СОКРАЩАЕМ ЗАВИСИМОСТИ

Благодаря удобному стандартному классу KDE нам не пришлось править код на предыдущем шаге. Надеюсь, это приободрит тех, кто еще не стал гуру программирования. Теперь нам предстоит сделать программу как можно более переносимой. По сути, это основная причина создания автопакета.

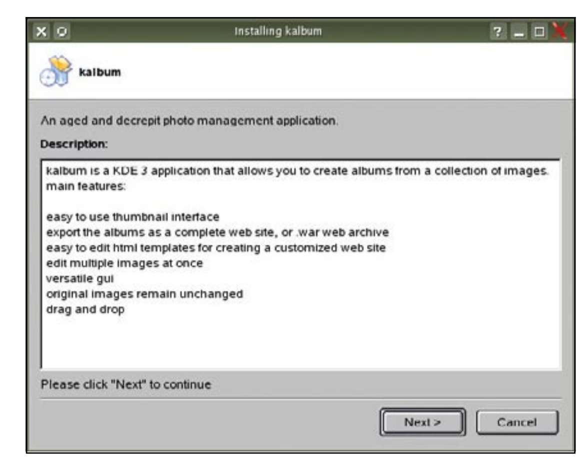
Главное препятствие к созданию повсеместно приемлемого пакета – огромное число зависимостей. Мы должны определить, какие из них действительно необходимы, а какие вполне можно отбросить. И это проблема с тройным дном. Любой разработчик прежде всего попытается найти и использовать стандартную библиотечную функцию, а не изобретать велосипед. Если программа использует всеобъемлющие API вроде *GTK* или *KDE/Qt*, то небольшой проект практически всегда обойдется функциями этих библиотек. Однако есть разработчики, предпочитающие выуживать библиотеки из самых потаенных омутов мира Open Source. Вот тут и начинаются проблемы с зависимостями.

Дело в том, что пользователю понадобятся не просто копии библиотек, а именно те версии, которые были у разработчика. Проблема усугубляется тем, что многие сборщики пакетов имеют привычку модифицировать библиотеки, накладывая на них патчи. В результате получается «ад зависимостей», сгубивший немало некогда добротных систем Linux.

Здесь нет простого решения. Если вы пишете программу и хотите, чтобы она была максимально переносимой, используйте поменьше сторонних библиотек, а те, что используете, подбирайте посвежее. Отсутствие старушки *foobarlib* 1997 года выпуска облегчит жизнь не только вам, но и создателю автопакета.

Внутреннее расследование

Хороший способ узнать, сколько библиотек тянется за программой – вызвать утилиту *ldd*. Сконфигурировав и построив пакет *Kalbum*, зайдите в каталог, где находится его исполняемый файл и введите команду: **ldd kalbum**. Вы увидите список из 38 библиотек, начинающийся так:



По завершении, приложение можно будет установить при помощи стандартного интерфейса пользователя.

```
linux-gate.so.1 => (0xffffe000)
libresolv.so.2 => /lib/libresolv.so.2 (0x4003c000)
libkio.so.4 => /opt/kde3/lib/libkio.so.4 (0x4004f000)
libkdeui.so.4 => /opt/kde3/lib/libkdeui.so.4 (0x4038c000)
libkwalletclient.so.1 => /opt/kde3/lib/libkwalletclient.so.1 (0x4067a000)
```

Легко понять, почему для KDE-программы требуются библиотеки *libkio* и *libkdeui*, но что такое, например, *linux*-gate.so.1? Если бы мы строили RPM из этого пакета как такового, RPM затребовал бы данную



На вид вывод *objdump* – бред, но без него не обойтись при поиске зависимостей.

<< библиотеку, а менеджер пакетов попытался бы ее установить, что вовлекло бы новые зависимости, и так далее. Но на самом деле *linux-gate* – экзотический случай: на нее ссылается вывод *ldd*, причем путь в ссылке не указан. И неудивительно: это не библиотека, а виртуальный динамический объект, используемый ядром для ускорения системных вызовов. Осталось убедиться, что *libkwalletclient* здесь лишняя, так как *Kalbum* не использует *Wallet Manager*, для связи с которым она и нужна.

Более подробную информацию по требуемым библиотекам можно добыть с помощью *objdump*. Команда **objdump -u kalbum** также выведет список библиотек, используемых программой. Библиотеки, расположенные в начале списка под заголовком *Dynamic Section*, помечены как 'NEEDED' (необходимые). Вас может огорчить тот факт, что список библиотек, выводимый *objdump*, не короче вывода *ldd*. Но ldd предусматривает еще один трюк: выводить названия библиотек, объявленных необходимыми, но никогда не использующихся:

```
ldd -u kalbum
Unused direct dependencies:
/lib/libresolv.so.2
/opt/kde3/lib/libkdesu.so.4
/opt/kde3/lib/libkwalletclient.so.1
```

Посмотрите на вывод – и увидите, что сюда попала и *libkwalletclient*. Итак, из 36 библиотек 31 не используется. Вопрос: как вычистить эти зависимости из программы? Разработчики *Autopackage* позаботились об этом и создали несколько Perl-скриптов для этой цели. Скрипты имеют общее имя *apbuild* и являются обертками для компилятора: они просто фильтруют ненужные зависимости перед тем, как пропустить необходимые.

Для компиляции *Kalbum* с использованием *apbuild* выполните **make CC=apgcc CXX=apg++** из главной директории приложения. Компилятор переадресуется на скрипты, а они попытаются минимизировать зависимости. Эффект от *apbuild* проявляется немедленно: список стал гораздо короче. По завершении компиляции проверьте получившийся файл командой **objdump –u kalbum**. Список используемых библиотек сократился с 36 до 9 – для KDE это мизер, и шансы создать переносимое приложение заметно повысились. С оставшимися зависимостями мы рассчитаемся на следующем шаге.

ШАГ 3 – СОБИРАЕМ АВТОПАКЕТ

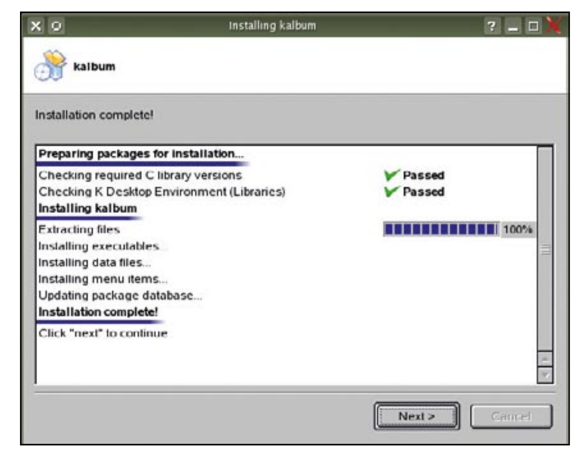
Радуйтесь: самое сложное позади. Все, что осталось сделать – создать небольшой текстовый файл, описывающий каждый компонент вашего пакета, с тем, чтобы *Autopackage* сумел собрать его в один инсталлируемый файл (к счастью, этот процесс в основном автоматизирован утилитой *makeinstaller*). Текстовый файл создается так, чтобы *makeinstaller* знал, что делать, и называется он spec-файлом. Если вы когда-либо собирали RPM-пакеты, вам известно, что они тоже используют файлы похожего формата. И фактически, и по названию это спецификация, включающая описание программы и перечень устанавливаемых файлов. Для автоматизации некоторых рутинных операций обычно используются скрипты, но кто-то, разбирающийся в приложении, должен отредактировать спецификацию вручную.

Метаданные

Первым делом изготовим спецификацию при помощи *makeinstaller*. Создайте в главном каталоге программы поддиректорию **autopackage**, именно в ней *Autopackage* будет искать spec-файл. Введите следующую команду для создания шаблона:

```
makeinstaller --mkspec >autopackage/default.apspec.in
```

Если вы откроете сгенерированный файл (**autopackage/default.apspec.in**), то обнаружите, что он состоит из разделов. Первый называется Meta и содержит описание программы, а также номер версии, чтобы *Autopackage* мог учесть его при обновлении. Назначение большинства параметров очевидно: легко догадаться, для чего нужны 'DisplayName', 'Maintainer' и 'Description'. 'ShortName' – сокращенное имя версии строчными буквами. Единственный параметр, способный вызвать затруднение – 'RootName'. Это уникальный идентификатор вашего автопакета, он начинается с @, затем следует домен, имя приложения и номер версии. Домен – не обязательно реальное имя, а просто символьный идентификатор: в нем требуется наличие хотя бы одного прямого слэша (но не в конце), а префиксы – ни www, ни протокол – не нужны. Номер версии – переменная, которую вставляет сам *Autopackage* ($SOFTWAREVERSION), так что ее можно оставить в покое. Для *Kalbum* этот параметр выглядит следующим образом:



Примерно такое вы увидите, если все прошло хорошо.

```
@paldandy.com/kalbum:$SOFTWAREVERSION
```

Последний трудный параметр, *AutoPackageTarget*, указывает версию *Autopackage*, необходимую для установки этого пакета. Мы используем версию 1.0, так и запишем (1.0).

Закончив править этот раздел, можете смело пропускать два следующих, BuildPrepare и BuildUnprepare. Они уже были сконфигурированы автоматически при создании шаблона. Это касается и раздела **import**, описывающего дополнительные файлы, которые нужно включить в пакет. После сборки, по умолчанию *Autopackage* проверяет каждый файл; если обнаружатся лишние файлы, их можно просто удалить.

Подготовка к запуску

Следующий раздел спецификации – самый важный. Он называется Prepare (Подготовка), и именно в нем перечисляются все зависимости программы. *Autopackage* предоставляет механизм установки большинства распространенных зависимостей, таких как KDE, Gnome, SDL, Perl, используя скелет-файлы. Они находятся в установочной директории

Autopackage (обычно **/usr/share/autopackage/skeletons**), и на них есть ссылка в спецификации, на базе описания имени домена, использованного как RootName автопакета.

При помощи списка зависимостей, составленного на предыдущем шаге, следует добавить запись о каждой библиотеке. Например, ваше приложение требует *libpng*, и в директории **skeleton** вы найдете скелет-файл, который (большинство из них этим и ограничивается) проверит наличие версии с требуемым номером. Если для нужной библиотеки скелет-файла найти не удалось, придется создать его самостоятельно, используя в качестве шаблона другие скелет-файлы.

Итак, после сборки *Kalbum* с помощью *apbuild*, у нас есть девять зависимостей: *libkio*, *libkdeui*, *libkdecore*, *libkdefx*, *libqt-mt*, *libXext*, *libX1 libstdc++* и *libc*. К счастью, все они поддерживаются *Autopackage*.

- Библиотеки *libk** устанавливаются вместе с KDE, так что в раздел Prepare нужно добавить такую строку: **'requireAtLeastVersion @ kde.org/kdelibs 3.0'.** 3.0 – самая ранняя версия KDE; для перестраховки можно указать номер поновее.
- *Libqt* требуется для KDE всегда, так что можно о ней не беспокоиться.
- Остальные четыре библиотеки есть в любом дистрибутиве Linux, тут тоже волноваться не о чем.

Так легко мы отделались благодаря сокращению числа библиотек до минимума на предыдущем шаге. Теперь раздел Prepare должна выглядеть следующим образом:

```
[Prepare]
# Dependency checking
require @kde.org/kdelibs 3.0
```

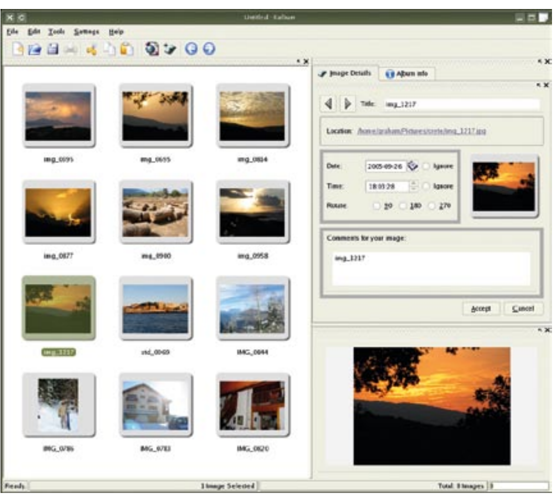
Следующий раздел называется Install. Здесь находятся несколько скриптов-макросов, которые устанавливают различные части вашего пакета. Например, *installExe* устанавливает исполняемый файл, а *installData* – требуемые программе файлы данных. *InstallDesktop* создает иконку на рабочем столе, а *InstallMan* – обеспечивает документацию. Исполняемых файлов у *Kalbum* только один, зато множество файлов с данными; секция *Install* для него выглядит так:

```
prefix=`kde-config --prefix`
installExe bin/*
installData share/apps/
installDesktop «Accessories» share/applications/kde/kalbum.desktop
```

Директории назначения самые обычные, как у других приложений. KDE-приложения требуют специального префикса для нормальной работы, который и добавляется в самом начале.

Итак, осталось только собрать сам автопакет. Поможет та же утилита *makeinstaller*, только теперь она используется немного по-другому:

```
makeinstaller autopackage/default.apspec.in
```



Разве не здорово? Теперь можно поделиться *Kalbum* с друзьями.

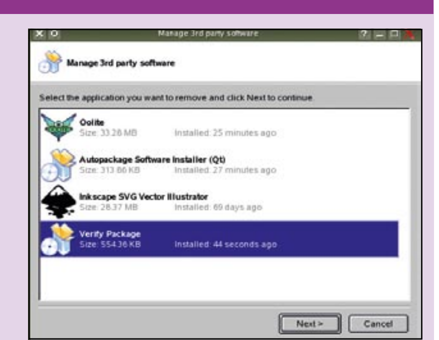
```
Building installer for kalbum...
```

Вот и все! В результате мы получим shell-скрипт с расширением. *autopackage*. Для установки программы нужно запустить его. Это и есть файл, которым вы должны сопроводить свое приложение. Дайте нам знать, как вы справились с этим заданием, если начнете делать свои автопакеты. LXF

ЧЕРЕЗ МЕСЯЦ
Извлечь все что можно из оболочки: разоблачим тайную жизнь команд. Они умеют больше, чем вы думаете!

КАК УДАЛИТЬ АВТОПАКЕТ

Простая установка – это, конечно, хорошо, но как потом избавиться от автопакета? Есть два пути. Во-первых, существует маленькая утилита, весьма похожая на диалог установки/удаления программ в Windows: просто наберите команду **autopackagemanager**. Она предоставляет интерфейсы для *Gtk* и *Qt*. Во-вторых, существует командная утилита package, предлагающая несколько опций для управления автопакетами. Например, для удаления *Kalbum* введите команду **package remove Kalbum**, а чтобы убедиться в правильности установки, замените **remove** на **verify**. Узнать больше о команде package вы можете в man- и info-документации.



Если вы не любите командные строки, автопакеты можно удалять через привычный графический интерфейс.

LINUX FORMAT ПРИНОСИТ ИЗВИНЕНИЯ

В прошлом номере мы по ошибке оставили вас без описания нового дистрибутива WINE. Исправляем эту трагическую ошибку.

WINE@ETERSOFT 0.9 LOCAL

Компания Этерсофт совместно с компанией Линуксцентр представляет продукт WINE@Etersoft 0.9 Local, предназначенный для запуска Windows-приложений в свободных операционных системах на платформе x86.

WINE@Etersoft позволяет использовать такие российские программы как 1С:Предприятие 7.7, 1С: Бухгалтерия 6.0, Консультант Плюс, Инфо-Бухгалтер, Кодекс, Референт, MS Office 97, программы подготовки обязательной отчётности и некоторые другие в операционных системах ALT Linux Master 2.4, ALT Linux 3.0 Compact, Debian 3.1 Sarge, FedoraCore 4, FreeBSD 5.4, Mandriva 2005, Mandriva 2006, MOPSLinux 3.x, Scientific Linux 4, Slackware 10.x, SuSe 10.

Если Вы ищете возможность сменить операционную систему, но желаете использовать Windows-приложения, то WINE@Etersoft – это решение для Вас!

В этой версии нет поддержки HASP-ключей для приложений 1С. Для того, чтобы иметь возможность работы с сетевыми версиями 1С, нужно приобрести Wine@Etersoft Network. Любопытно то, что на сайте Etersoft есть инструкции для запуска 1С на обычном Wine. Однако если вам нужна техническая поддержка и стабильная работа программ 1С и Консультант Плюс, рекомендуем вам приобрести этот пакет.





ПРОГРАММИРОВАНИЕ

Уроки Python

ЧАСТЬ 3 Итак, познакомившись с типами данных, доступных в языке *Python*, мы, под чутким руководством **Сергея Супрунова** морально готовы приступить к более серьёзным вещам.
На сегодня программа-минимум – разобраться с модулями и научиться писать свои функции.

МЕСЯЦ НАЗАД

Мы изучали типы данных Python

Модули

Начнем со стандартных модулей. Стандартными их называют потому, что они, как правило, уже включены в вашу поставку *Python* (будь то Linux, FreeBSD или даже Windows) и не требуют дополнительной инсталляции. Как вы видели в конце прошлого урока, модуль подключается оператором import. В качестве параметров вы можете указать любое количество модулей. Также допускается любое число операторов **import** в любой точке вашей программы.

Для чего же нужны модули? Чуть позже, когда мы напишем свою функцию, вы увидите, что с помощью модуля очень легко реализуется концепция повторного использования кода. Пока же модули интересны нам как способ расширения возможностей вашей программы. С их помощью вы подключаете уже написанные кем-то наборы функций и методов, которые сможете использовать почти так же, как если бы они были встроенными (такими, как рассмотренная нами ранее **float()**).

Начнем с модуля **sys**. В нем определен ряд переменных, содержащих информацию об операционной системе, дескрипторы стандартных потоков ввода, вывода и ошибок (**stdin**, **stdout** и **stderr** соответственно), и так далее. Рассмотрим небольшой пример:

```
>>> import sys
>>> print sys.arch, sys.platform
i386 linux2
>>> print sys.version
2.4.1 (#1, Sep 13 2005, 00:39:20)
[GCC 4.0.2 20050901 (prerelease) (SUSE Linux)]
>>> print sys.maxint
2147483647
```

Как видите, чтобы получить доступ к определённым в модуле переменным (впрочем, это относится и к остальным объектам), перед их именем следует указать имя модуля. Это оберегает от конфликта имён, когда объявленная вами переменная или функция имеет такое же имя, как и определенная в модуле. То есть каждый модуль имеет свое пространство имен, доступ к которому осуществляется по имени модуля. Если имя модуля слишком неудобно для работы, вы можете указать для него псевдоним после ключевого слова **as** (например, **import SimpleXMLRPCServer as rpc**), и в дальнейшем использовать это упрощённое имя. Впрочем, иногда удобнее (а порой и необходимо), чтобы имена, определенные в модуле, были импортированы непосредственно в пространство имён основной программы. Для этого предусмотрена несколько иная конструкция:

```
>>> from sys import *
>>> print platform
linux2
```

Теперь можно обращаться к объектам модуля непосредственно, но будьте осторожны – вы потеряете все определённые вами переменные и функции, имена которых совпадут с именами определенных в модуле.

Символ «звездочка» в операторе **from** - **import** означает, что должно быть импортировано всё, что определено в модуле. Вы также можете перечислить только то, что желаете сделать доступным в вашей программе (например, **from sys import version**).

Ещё одна важнейшая переменная модуля sys – список sys.argv, в который в момент запуска *Python*-программы помещаются аргументы, переданные в командной строке. Пример использования этого списка вы увидите далее в данной статье.

Модуль os

Пожалуй, самый популярный модуль *Python* – это **os**. Он предоставляет вам ряд функций, позволяющих взаимодействовать с операционной системой: **chroot()**, **mkdir()**, **setuid()** и так далее. Названия, думаю, говорят сами за себя. А как посмотреть полный список функций, доступных в модуле, я надеюсь, вы ещё помните с прошлого урока: **dir(os)**.

Для работы с файловой системой практически незаменим подмодуль этого модуля – **os.path**. С его помощью вы можете определять типы файлов, установленные для них права доступа, и тому подобное. Нам ещё доведётся поработать с ним на практике.

Прочие модули

Вообще, в стандартную поставку *Python* входит несколько десятков наиболее полезных модулей. Найти их можно в каталоге **/usr/lib/python** (на некоторых системах путь может быть иным, например, **/usr/local/lib/python)**. Если быть точнее, то *Python* ищет свои модули в каталогах, список которых возвращает переменная **sys.path**.

Обращайте внимание на файлы с расширением **py** (**pyc** и **pyo** – это байт-код модулей, который используется для экономии ресурсов, поскольку при обращении к модулю повторная компиляция не требуется). Названия большинства из них говорят сами за себя, к тому же вы можете открыть любой модуль, чтобы ознакомиться с его кодом и, как правило, очень подробными комментариями. С некоторыми из них мы обязательно столкнёмся в дальнейшем.

Функции

Со встроенными функциями вы уже знакомы. Но что делать, если возможностей, заложенных в интерпретатор или стандартные модули, вам недостаточно? Всё просто – напишите собственную функцию!

Пользовательская функция определяется оператором **def** (от **define** – определять). В качестве параметра указывается имя функции (оно должно быть уникально в пределах пространства имён основной программы), а в скобках перечисляются её формальные аргументы. Далее ставится двоеточие (как и в случае любых блочных операторов, таких как **if** или **for**), и на последующих строках, имеющих отступ, вводится код функции. Впрочем, когда речь идёт о *Python*, несколько строк кода способны сказать больше, чем пол-страницы текста на естественном языке. Чтобы лучше во всем разобраться, рассмотрим один пример (файл назовем **abbr.py**):

```
i#!/usr/bin/python
# -*- coding: utf-8 -*-
“””
```

Программа обслуживает небольшой словарь сокращений, используя **anydbm**

```
“””
import sys, anydbm as db
dictfile = ‘dict.dbm’
def openDictionary(dbmfile=dictfile):
    “”” Открыть словарь “””
    global dictionary
    try:
        dictionary = db.open(dbmfile, ‘c’)
    except:
        raise ‘Ошибка открытия файла’
def getMeaning(abbr):
    “”” Поиск расшифровки “””
    try:
        return dictionary[abbr]
    except:
        return ‘[не определено]’
def addMeaning(abbr, meaning):
    “”” Добавить сокращение “””
    dictionary[abbr] = meaning
  def delMeaning(abbr):
    “”” Удалить сокращение “””
    try:
        del dictionary[abbr]
    except:
        pass
if __name__ == “__main__”:
    openDictionary()
    try:
        command = sys.argv[1]
        if command == ‘-a’:
            addMeaning(sys.argv[2], sys.argv[3])
        elif command == ‘-d’:
            delMeaning(sys.argv[2])
        else:
            abbr = sys.argv[1]
            print abbr + ‘ = ‘ + getMeaning(abbr)
    except:
        print “””\
```

Неверные аргументы. Допустимый синтаксис:

```
abbr.py <ABBR>                - расшифровать сокращение
abbr.py -a <ABBR> <MEANING>  - добавить расшифровку
abbr.py -d <ABBR>             - удалить сокращение
“””
else:
    openDictionary()
```

Эта программа реализует простейший консольный словарь сокращений. Только не забудьте сделать файл исполняемым (**chmod u+x abbr.py**). Поддерживаются три операции (информацию об этом вы получите, если попытаетесь вызвать сценарий с неправильными аргументами):

1) abbr.py <ABBR> – возвращает из словаря расшифровку сокращения <ABBR>;

2) abbr.py -a <ABBR> <MEANING> – добавляет в словарь расшифровку сокращения. В силу особенностей словарного типа данных в *Python*, эта же команда изменяет расшифровку для уже существующей аббревиатуры;

3) abbr.py -d <ABBR> – удаляет из словаря запись для указанного сокращения.

Обратите внимание на вторую строку – здесь указывается кодировка исходного текста (подробности см. на странице **http://www.python.org/peps/pep-0263.html**). Если вы используете не UTF-8, то укажите в этой строке свою кодировку (например, koi8-r).

Строки, обрамлённые утроенными кавычками, в данном случае можно рассматривать как многострочные комментарии.

Оператор **global** в функции **openDictionary()** делает переменную **dictionary** доступной за пределами функции. В противном случае её действие распространялось бы только на эту функцию. Ещё один интересный момент – то, как описан формальный аргумент функции. В данном случае мы задаем для переменной значение по умолчанию. Если в дальнейшем при вызове функции вы не передадите в неё аргумент, будет использовано значение переменной **dictfile**.

Несколько слов нужно сказать о самом модуле **anydbm**. С его помощью *Python* выполняет поиск доступного dbm-пакета в текущей системе (вам не нужно об этом беспокоиться). Данные в файле DBM организованы по словарному принципу (ключ – значение), поэтому данный файл может быть естественным образом сопоставлен с переменной типа «словарь». Все дальнейшие операции с данными словаря для программиста выглядят как обычные, а **dbm**-модуль позаботится о том, чтобы все данные оказались сохранены в файл. Собственно говоря,

ПОЛЕЗНЫЙ СОВЕТ

Когда вы работаете в интерактивном терминале, к вашим услугам всегда функция **help()**. Используйте её, когда вам понадобится подсказка по той или иной функции или модулю. Не забывайте заключать в кавычки её аргумент – ведь вы передаёте имя функции, а не ее саму: **help(‘dir’)**, **help(‘sys’**.

dbm-файл вы можете рассматривать как словарь, содержимое которого не теряется между запусками программы. Также обратите внимание, что при подключении этого модуля мы используем псевдоним.

Итак, в функции **openDictionary()** мы открываем файл **dict.dbm** (второй параметр – '**c**' – даёт указание создать такой файл, если его ещё нет) и сопоставляем его со словарём **dictionary**. Конструкция «**try – except**» служит для обработки ошибок. В блок **try** мы помещаем потенциально опасные операции (которые могут завершиться с ошибкой), а блок **except** содержит код, который будет выполнен в случае возникновения ошибки в блоке **try**. Оператор raise генерирует сообщение об ошибке и прерывает выполнение программы.

Далее описаны три функции, реализующие заявленную выше функциональность. Пожалуй, в пояснении нуждается здесь только оператор pass в функции **delMeaning()**: это пустой оператор, который ничего не делает – если нужного ключа в словаре нет, то и удалять нечего.

Ну и в конце сценария, в секции оператора **if**, вы можете увидеть пример обработки аргументов командной строки.

Конечно, этот учебный пример не лишён недостатков: невозможно сделать несколько записей, имеющих одинаковую аббревиатуру, работа чувствительна к регистру символов (введя запись для «UDP», вы не сможете получить значение по ключу «udp»), и т.д. Однако сейчас нам важен не функционал, а собственно языковые конструкции.

Возвращаясь к модулям

Итак, мы написали функцию, которая может оказаться крайне полезной в других программах. Как бы было хорошо, если бы оказалось возможным поместить её в модуль! А ведь нет ничего проще – она уже в модуле. Да-да! Вы можете просто подключить написанную выше программу в качестве модуля (одним из операторов, **import** или **from** – **import**) и спокойно использовать определённые в ней переменные и функции!

В этом заключается одно из преимуществ *Python* по сравнению с его «прямым конкурентом» - языком Perl. В Perl вам пришлось бы специально оформлять вашу программу как модуль (как минимум, потребуется оператор **package**). В Python концепция повторного использования кода заложена в основу синтаксиса и не требует от вас совершенно никаких усилий.

Рассмотрим пример:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
import abbr
abbr.delMeaning('UDP')
print abbr.getMeaning('IP')
```

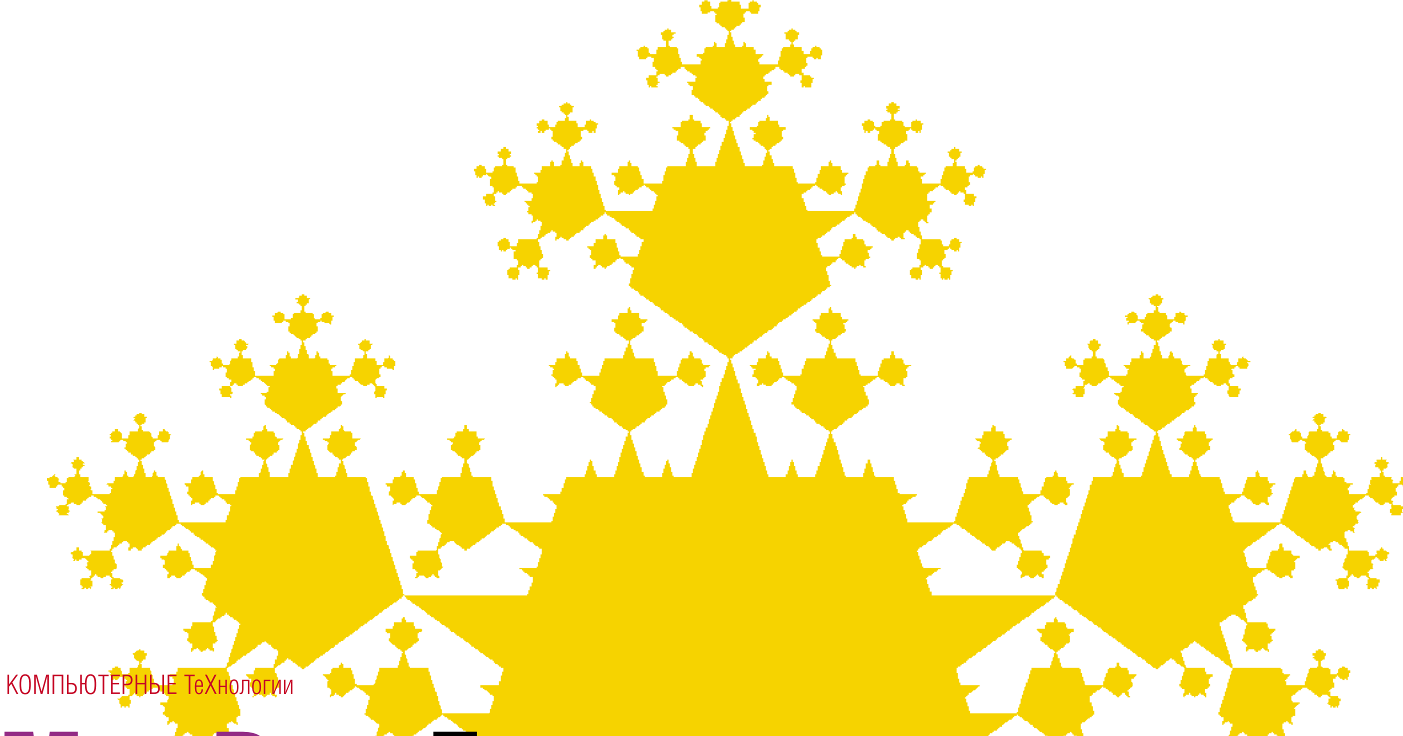
Как видите, вам достаточно импортировать **py**-программу (расширение не указывается), и вы можете в полной мере воспользоваться всем созданным ранее. Единственное, этот файл должен быть доступен для поиска (самый простой вариант – поместить его в текущий каталог).

Обратите внимание, что в момент импортирования переменная **__name__** в подключаемом модуле будет иметь значение, совпадающее с именем модуля. Поэтому в данном случае вместо секции **if** сценария **abbr.py** будет выполнена секция **else**. Подобный оператор **if – else** присутствует практически в каждой программе на языке Python и позволяет использовать ее и как модуль, и как самостоятельный сценарий (в последнем случае, как нетрудно догадаться, переменная **__name__** будет иметь значение "**__main__**").

Итак, вы уже в состоянии разрабатывать довольно серьёзные и полезные программы. Однако не научившись работать с классами, вы не познаете всю мощь *Python*. В следующий раз мы заполним и этот пробел.







КОМПЬЮТЕРНЫЕ ТеХнологии

MetaPost Базовые элементы

Определённо всё из чего-то состоит. Элементарные кирпичики – вот что интересно. **Евгений Балдин** продолжает уроки рисования.

Объяснять компьютеру что Вы хотите сделать гораздо сложнее, чем нарисовать самому. Компьютеру надо объяснять абсолютно всё – телепатические способности у машин на текущий момент отсутствуют. Но, объяснив один раз, все похожие действия выполняются путём небольшой модификации уже готовых инструкций. В результате, потраченное на объяснение время себя полностью оправдывает, правда, при этом требуются дополнительные «мозговые усилия».

Рисуем по точкам

Первое, что надо сделать перед созданием нового рисунка, это сделать его набросок на миллиметровке. Допустим, необходимо нарисовать черепашку:

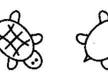
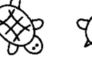
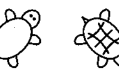
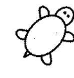
Делаем это, как можем, а затем очень подробно объясняем компьютеру, что мы хотим от него. Инструкции очень простые:

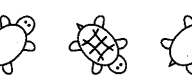
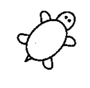
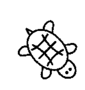
```
%Файл coord.mp
%Черепашка
beginfig(1) ;
 numeric u;u:=5mm;
 draw (-5u,0u){dir 90}..{dir 0}(0,3.5u){dir0}..
   {dir -90}(5u,0){dir -90}..{dir 180}(0u,-3.5u){dir 180}..
   {dir 90}cycle withpen pencircle scaled 0.4u;
 draw (4.5u,1.5u)..(7u,1.5u)..(8u,0u)..(7u,-1.5u)..
   (4.5u,-1.5u) withpen pencircle scaled 0.4u;
 draw (6.5u,.5u) withpen pencircle scaled 0.8u;
 draw (6.5u,-.5u) withpen pencircle scaled 0.8u;
 draw (-3.5u,-2.5u)..(-4.2u,-4.5u)..(-3.7u,-5u)..
   (-2.4u,-4u)..(-2u,-3.5u) withpen pencircle scaled 0.4u;
 draw (-3.5u,2.5u)..(-4.2u,4.5u)..(-3.7u,5u)..(-2.4u,4u)..
   (-2u,3.5u) withpen pencircle scaled 0.4u;
 draw (3.5u,-2.5u)..(4.2u,-4.5u)..(3.7u,-5u)..(2.4u,-4u)..
   (2u,-3.5u) withpen pencircle scaled 0.4u;
 draw (3.5u,2.5u)..(4.2u,4.5u)..(3.7u,5u)..(2.4u,4u)..
   (2u,3.5u) withpen pencircle scaled 0.4u;
 draw (-5u,0.5u)--(-6.5u,0)--(-5u,-0.5u)
   withpen pencircle scaled 0.4u;
 draw (0u,2.5u)..(0,0u)..(0u,-2.5u)
   withpen pencircle scaled 0.3u;
 draw (-2.4u,2.4u)..(-2u,0u)..(-2.4u,-2.4u)
   withpen pencircle scaled 0.3u;
 draw (2.4u,2.4u)..(2u,0u)..(2.4u,-2.4u)
   withpen pencircle scaled 0.3u;
 draw (-3u,1.5u)..(0,1u)..(3u,1.5u)
   withpen pencircle scaled 0.3u;
 draw (-3u,-1.5u)..(0,-1u)..(3u,-1.5u)
   withpen pencircle scaled 0.3u;
endfig ;
```

Рисуем по точкам, используя команду draw. Каждая точка задаётся парой чисел «(x,y)» – x и y координаты соответственно. Точки соединяются либо прямыми линиями «--», либо кривыми «..». Кривые, соединяющие точки описываются полиномом Бернштейна третьей степени (*Сергей Николаевич Бернштейн 1912*). Часто их называют кубическими кривыми Безье (*Piere Bezier 1960*).

Компьютер обязательно должен знать каким пером (withpen pencircle) и какой толщины (scaled 0.3u) рисуется текущая линия. Такие объяснения могут показаться избыточными, но всегда можно скопировать предыдущую инструкцию и поправить её под свои нужды, а многословность облегчает чтение кода.

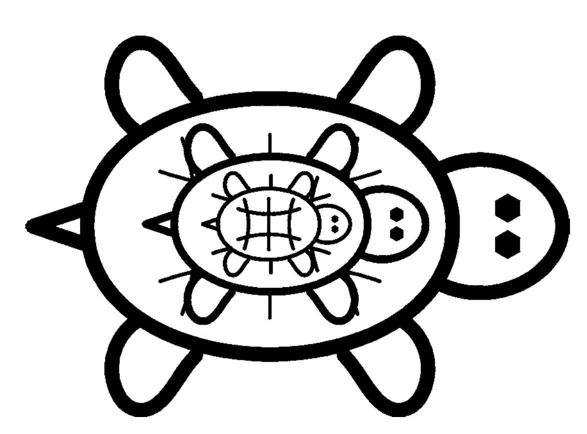
Далее этот рисунок можно использовать многократно, например, для составления какого-либо узора:

Обратите внимание, что почти все числовые значения представляют из себя коэффициент умноженный на параметр, например, «2.4u». Параметру «u» можно присвоить числовое значение в миллиметрах (параметр «mm»). Есть несколько определённых по умолчанию значений длины, например, «cm» соответствует сантиметру.

При программировании на *Meta* по мере возможности используйте параметрические зависимости. Нет необходимости «прибивать что-то гвоздями». В данном случае наличие параметра позволяет легко изменить масштаб рисунка:



Очевидно, что в предложенном решении одного параметра мало, так как «u» контролирует не только геометрические размеры, но и размеры пера.

Точки, также можно представить как переменные, имеющие тип pair:

```
numeric u;
u:=0.5mm;
pair A,B;
A:=(1u,2u);B=(5u,10u);
draw A--B withpen pencircle scaled 0.3u;
```

Следует обратить внимание, что знак присвоения «:=» в случае переменной A и знак равенства в случае переменной B здесь действует одинаково. Отличия возникают, когда используется уникальная способность Meta воспринимать и решать систему линейных уравнений. В дальнейшем в объяснениях буквы A и B будут использоваться как переменные типа точка.

Meta позволяет создавать свои типы перьев, но для рисования простых рисунков использовать что-то отличное от круглого пера (*pencircle*) нет особой необходимости. Если вам хочется расширить ваши познания в этой области, то обратитесь к любому руководству по *MetaFont* или *MetaPost*. Нет необходимости каждый раз указывать каким пером следует рисовать. Достаточно выбрать какое-либо перо по умолчанию с помощью команды pickup, например, так:

```
pickup pencircle scaled 0.2u;
```

Пути

Инструкции draw позволяет рисовать сплошную линию. На её основе в *MetaPost* создана команда для рисования стрелки drawarrow. Воспользуемся ей для изображения суммы векторов (рис. справа)



Кроме непосредственно команды draw, в этом рисунке необходимо использовать команду для вставки текстовых меток label, которая будет подробно разобрана позже:

```
%Файл path.mp
%Закон Ньютона - векторная сумма сил
beginfig(1);
numeric u;
u := 1mm;
drawarrow (0,0)--(20u,30u) withpen pencircle scaled 0.3u;
label.ulft(btex \(\vec{F}_1\) etex,1/2[(0,0),(20u,30u)]);
drawarrow (20u,30u)--(40u,30u) withpen pencircle scaled 0.3u;
label.top(btex \(\vec{F}_2\) etex,1/2[(20u,30u),(40u,30u)]);
drawarrow (40u,30u)--(50u,10u) withpen pencircle scaled 0.3u;
label.urt(btex \(\vec{F}_3\) etex,1/2[(40u,30u),(50u,10u)]);
drawarrow (50u,10u)--(35u,15u) withpen pencircle scaled 0.3u;
label.llft(btex \(\vec{F}_4\) etex,1/2[(50u,10u),(35u,15u)]);
drawarrow (0u,0u)--(35u,15u) withpen pencircle scaled 0.8u;
drawarrow (0u,0u)--(35u,15u) withpen pencircle scaled 0.3u
  withcolor white;
draw (0u,0u)  withpen pencircle scaled 2u ;
label.bot(btex \(\vec{F}_1+\vec{F}_2+\vec{F}_3+\vec{F}_4\) etex,
  1/2[(35u,15u),(0u,0u)]) rotatedaround
            (1/2[(35u,15u),(0u,0u)],angle(35,15));
endfig;
```

Конструкция вида «1/2[A,B]» имеет тип pair и равна точке, расположенной ровно по середине между точками A и B. Точно так же можно выбрать точку на линии AB, но делящую эту линию в отношении 1 к 2: «1/3[A,B]», или 1 к 4: «0.2[A,B]»

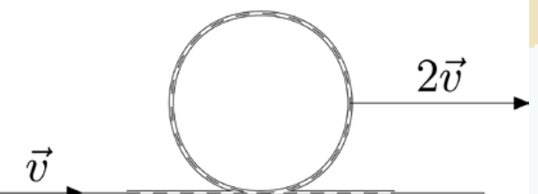
Кроме drawarrow в *MetaPost* определена команда drawdblarrow, которая рисует кончик стрелки на обоих концах пути.

Команды, типа draw, работают с объектами path (путь). Путь – это набор из точек (тип pair) с описанием того, как эти точки соединяются друг с другом. Минимальный путь – это одна точка. При рисовании такого пути на рисунке остаётся отпечаток в форме пера. Во всех приведённых здесь примерах перо имеет круглую форму *pencircle*, поэтому в этом случае возникает просто точка. Часто бывает удобно создать переменную типа path:

```
%Файл path.mp
%Рис к 1.6.8 б) нить пропущенная через изогнутую трубу
beginfig(2) ;
 numeric u;
 u = 0.8mm;
 path p;
 p:=(5u,0u)--(20u,0u){dir 0}..(20u,20u)..
             {dir 0}(20u,0u)--(35u,0u);
 cutdraw p  withpen pencircle scaled 1.5u;
 draw p withpen pencircle scaled 1u withcolor white;
 draw p withpen pencircle scaled 0.3u
             dashed evenly scaled 1/2u;
 drawarrow (-10u,0u)--(0u,0u)
            withpen pencircle scaled 0.3u;
 draw (-10u,0u) withpen pencircle scaled 1u;
 draw (-10u,0u)--(5u,0u) withpen pencircle scaled 0.3u;
 drawarrow (30u,10u)--(50u,10u)
            withpen pencircle scaled 0.3u;
 draw (30u,10u) withpen pencircle scaled 1u;
```

```
  draw (35u,0u)--(45u,0u) withpen pencircle scaled 0.3u;
  label.top(btex \(\vec{v}\) etex,(-5u,0u));
  label.top(btex \(2\vec{v}\) etex,(40u,10u));
endfig;
```

В данном примере требовалось изобразить трубу изогнутую буквой «**0**», сквозь которую продета нить. Для этого определяем путь «**p**». Затем этот путь используется в трёх командах как переменная: рисуется толстая чёрная труба шириной «**1.5u**», внутри неё рисуется более тонкая шириной «**1u**» белого цвета (withcolor white) – получается полая труба, а затем внутри трубы отрисовывается нить, причём нить рисуется пунктиром (dashed evenly).



Первый и последний участок изогнутой трубы прямые линии, поэтому соединение с первой и последней точкой описывается как «**--**». Чтобы нарисовать кривую, похожую на круг, достаточно двух точек:

```
numeric u,R; u=1mm;R=10u;
pair A,B; A:=(-R,0);B:=(0,R);
path P; P:=A..B..cycle;
draw P withpen pencircle 1u;
```

В этом примере определяются две точки A и B. Путь P строится по этим точкам, при этом выходя из точки A, мы попадаем в точку B, а затем снова в точку A: команда cycle позволяет создавать замкнутую кривую. Настройки *Meta* по умолчанию таковы, что получившаяся кривая достаточно хорошо совпадает с точной окружностью. Чтобы лучше совпадать с окружностью нужно больше «опорных» точек. Для обычных рисунков хватает точности построения по двум точкам, но всё-таки надо осознавать, что отличие есть.

Для уточнения пути в некоторых точках можно указать под каким углом должна подходить кривая к этой точке. Инструкция вида «dir α » если она находится перед точкой, указывает под каким углом кривая должна подходить, а после точки – под каким углом кривая должна уходить. α – угол в градусах от оси абсцисс.

Для указания направления можно так же воспользоваться сокращениями left, right, up и down, которые означают, соответственно, влево, вправо, вверх и вниз.

Существует несколько типов соединений между точками. Два из них «..» и «--» мы уже изучили. Из других типов полезен тип «&» – «сращивание» (объединение двух путей без влияния друг на друга) и «---» – «натянутая» линия (то же, что и «--», но влияет на соседние соединительные участки).

Обратите внимания на команду cutdraw. Так как *MetaPost* рисует с помощью перьев, то в случае круглого пера (pencircle), концы линий также округлые. В случае, когда необходимо «обрубить» концы, избавиться от округлостей на концах линий, и используется эта инструкция. Следует отметить, что эта команда «подчистки» концов использует другую более общую команду

```
cutoff(«точка»,«угол»);
```

Вставка текста

MetaFont был создан как специализированный инструмент для создания шрифтов. *MetaPost* создавался как инструмент для рисования любых изображений, поэтому в него был встроен довольно мощный механизм вставки текста. Работая совместно с *LaTeX*, *MetaPost* позволяет использовать всю мощь текстового процессора для создания надписей. Кроме всего прочего, иметь в рисунках те же шрифты, что и в тексте просто красиво.

Разберём это на примере треугольник Паскаля – треугольной числовой таблицы для составления биномиальных коэффициентов. По боковым сторонам треугольника стоят единицы, внутри треугольника числа

Треугольник Паскаля

$(a+b)^n = \sum C_n^k a^k b^{n-k}$

```
0:  1
1:  1 1
2:  1 2 1
3:  1 3 3 1
4:  1 4 6 4 1
5:  1 5 10 10 5 1
6:  1 6 15 20 15 6 1
7:  1 7 21 35 35 21 7 1
8:  1 8 28 56 70 56 28 8 1
9:  1 9 36 84 126 126 84 36 9 1
10: 1 10 45 120 210 252 210 120 45 10 1
11: 1 11 55 165 330 462 462 330 165 55 11 1
12: 1 12 66 220 495 792 924 792 495 220 66 12 1
13: 1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
14: 1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
```

образуются сложением двух чисел, стоящих над данным::

Так как нижние строки из-за увеличения разрядности чисел расползались, то для представления появилась необходимость уменьшить их геометрические размеры. В коде так же использовались циклы, которые будут разобраны позже. Команда show позволяет отлаживать код, выдавая значения переменных на экран во время компиляции рисунка с помощью **mpost**.

```
%Файл pic.mp
%Треугольник Паскаля
beginfig(5) ;
 numeric u;
 u = 1.mm;
 numeric dy,dx,x,y,n[ ][ ],i,j,sy,ds,nlast;
 dy:=5u;dx:=5u;x=0;y=0;
 ds=0.04;sy=0.032;nlast=14;
 picture z;
 for i:=0 upto nlast:
   dy:=dy*(1-sy);
   y:=y-dy;
   for j:=0 upto i:
     if (j=0) or (j=i):
             n[i][j]:=1;
     else:
             n[i][j]:=n[i-1][j-1]+n[i-1][j];
     fi
%      show i,j,n[i][j];
     z:=thelabel(decimal(n[i][j]),(0,0));
     x:=dx*(j-i/2);
     label(z scaled (1-ds*i),(x,y));
   endfor
   z:=thelabel.lft(decimal(i)&":",(0,0));
   label(z scaled (1-ds*i),(dx*(-nlast/2-1),y));
 endfor
 label.rt(btex \((a+b)^n=\) etex,(5u,-5u));
 label.rt(btex \(=\sum C^k_na^kb^{n-k}\) etex,
                    (10u,-10u));
 label.lft(btex Треугольник Паскаля etex
                    rotated 56,(-5u,-20u));
endfig ;
```

Команда decimal делает из переменной типа numeric строку. Две строки можно слить с помощью операнда «&».

Для того чтобы вставить текстовую метку в рисунок, необходимо получить на входе текстовую строку, которую, возможно, надо будет преобразовать с помощью LaTeX, и точку где эту строку следует расположить:

```
label("text string",A);
```

Полезно ещё уточнить с какой стороны от указанной точки расположить текстовую метку. Уточнение производится с помощью суффикса, который добавляется макросу label через точку. Всего существует восемь стандартных суффиксов: «.rt» – расположить справа, «.lft» – слева, «.top» – сверху, «.bot» – снизу, «.llft» – расположить снизу и

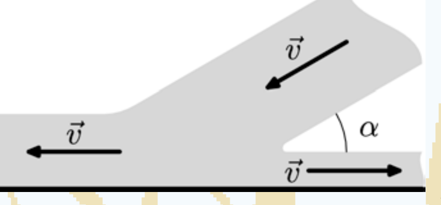
слева по диагонали, «.lrt» – снизу и справа, «.ulft» – сверху и слева, «.urt» – сверху и справа.

Если label передаётся просто строка (тип string), то текст обрабатывается силами *MetaPost* и всё, что выходит за пределы ASCII-таблицы с большой вероятностью не отобразится. Для того чтобы строка была обработана *LaTeX*, необходимо указать это с помощью разделителей btex и etex. Строка между этими разделителями обрабатывается *LaTeX*, при этом в качестве заголовка используются инструкции перечисленные в начале mp-файла между verbatimtex и etex. Таким образом можно использовать кириллицу и любую конструкцию, которую понимает *LaTeX*.

В некоторых случаях команде label удобно передавать не строку, а картинку picture. В нашем случае это было необходимо, так как надпись надо было масштабировать. Объект picture представляет из себя совокупность примитивов типа путей и точек, поэтому его можно трансформировать. Команда thelabel создаёт такую картинку. Объект, заключённый между btex и etex так же является картинкой.

Заливка

Кроме рисования кривых часто бывает необходимо закрасить какую-либо замкнутую область.



Для этого существует команда fill. На вход команды fill подаётся объект типа path, при этом путь должен быть замкнутым, то есть оканчиваться командой cycle. Команда cycle автоматически замыкает кривую.

```
%Файл pic.mp
%Рис к 1.6.20 а) струя под углом, разбивающаяся о пол
beginfig(10) ;
  numeric u;
  u = 1mm;
  numeric st;st:=u/(sqrt 3);
  path p;
  p:=(0,10u)--(18u,10u){dir 0}..{dir 30}(22u,10u+2st)--
     (50u,10u+30st){dir -90}..{dir -90}(60u,5u+20st)--
     (42u,5u+2st){dir -150}..{dir 0}(42u,5u)--
     (60u,5u){dir -120}..{dir -120}(60u,0u)--
     (0u,0u){dir 60}..{dir 60}cycle;
  fill p  withcolor 0.8 white;
  draw ((40u,5u)+10u*dir 30){dir -60}..{dir -90}(50u,5u);
  label(btex \(\alpha\) etex,(53u,8u));
  draw (0u,0)--(60u,0u) withpen pencircle scaled 0.8u;
  numeric R,alpha;R:=12u;alpha=30;
  drawarrow (25u+25u,5u+25st)--
               ((25u+25u,5u+25st)-R*dir alpha)
                    withpen pencircle scaled 0.6u;
  label.ulft(btex \(\vec{v}\) etex,1/2[(25u+25u,5u+25st),
   ((25u+25u,5u+25st)-R*dir alpha)]);
  drawarrow (20u,5u)--(20u-R,5u)
                    withpen pencircle scaled 0.6u;
  label.top(btex \(\vec{v}\) etex,(20u-R/2,5u));
  drawarrow (45u,2.5u)--(45u+R,2.5u)
                    withpen pencircle scaled 0.6u;
  label.lft(btex \(\vec{v}\) etex,(45u,2.5u));
endfig ;
```

Для fill существует противоположная по смыслу команда unfill, которая, соответственно, убирает заливку в выбранной окрестности. Если вы планируете использовать эти команды, то изучите соответствующий раздел «Всё про METAFONT» Кнута, так как влияние этих команд друг на друга не совсем тривиально. Дело в том, что при наложении друг на друга двух заливок в месте пересечения образуется «двойной слой краски» и для того чтобы убрать его необходимо дважды вызывать команду unfill. Команда unfill действует схоже, только число «слоёв краски» становится отрицательным. Для сведения всё к ситуации когда краска либо есть (один слой), либо её нет используется команда «выравнивания» cullit.

Обратите внимание, что точку можно задать не только парой чисел: «**(x,y)**» – Декартова система координат, но и радиусом с направлением «**R*dir** α » – полярные координаты.

Цвета

Ещё одно важное отличие *MetaPost* от *MetaFont* – это наличие цвета. Работа с цветом обычно сложнее, чем работа с чёрно-белым рисунком. До сих пор процедура переноса электронного цветного рисунка на бумагу не является тривиальной. Но время идёт, и цветные принтеры становятся всё доступней. Кроме того, хорошие чисто электронные тексты, не привязанные к твёрдой копии, так же становятся весьма распространёнными. Поэтому если необходимо, то следует пользоваться цветом, естественно, если вы знаете что делать. В случае простого рисунка цвет, как правило, только отвлекает.

В *MetaPost* управление цветом реализовано на довольно низком уровне. Цвет определяется объектом типа color и представляет из себя тройку чисел принимающих значение от 0 до 1: «**(r,g,b)**», где **r** соответствует красной, **g** – зелёной, а **b** – голубой компоненте. Существует пять предопределённых цветовых констант: red **(1,0,0)**, green **(0,1,0)**, blue **(0,0,1)**, а так же black **(1,1,1)** и white **(0,0,0)**.

Нарисовать объект, выбранным цветом, можно с помощью инструкции withcolor за которой следует сам цвет. Цвета можно складывать, вычитать, умножать на число. Пользуясь базовыми определениями можно создать более сложные объекта. Например как этот спектр:

Код, создающий эту картинку, далёк от совершенства, но он прост, и его можно улучшать в случае необходимости.

```
%Файл colors.mp
def spectrline(expr ic,c,l,w,dw) =
  begingroup
    save i,ifirst,ilast,istep,icc;
    color icc;
    if (ic>0):istep:=1;ifirst:=0;ilast=255;
    else:istep:=-1;ifirst=0;ilast:=-255; fi;
    if (abs(ic)=1):icc:=red;fi;
    if (abs(ic)=2):icc:=green;fi;
    if (abs(ic)=3):icc:=blue;fi;
    for i:=ifirst step istep until ilast:
      draw ((0u,0u)--(0u,l)) shifted (w+(dw/2*abs(i),0))
      withpen pencircle scaled dw withcolor (c+i/255*icc);
    endfor;
  endgroup;
enddef;
%спектр
beginfig(1) ;
  color current;
  u:=1mm;w:=200u;dw:=w/512;l:=5u;
  current:=(1,0,0);
  spectrline(2,current,l,(1/4w,0u),dw)
  current:=(1,1,0);
  spectrline(-1,current,l,(1/2w,0u),dw)
  current:=(0,1,0);
  spectrline(3,current,l,(3/4w,0u),dw)
  current:=(0,1,1);
  spectrline(-2,current,l,(w,0u),dw)
endfig ;
```

В примере используется макрос spectrline для уменьшения размера кода.



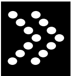
ЧЕРЕЗ МЕСЯЦ

Мы займемся автоматизацией и постараемся переложить на MetaPost большую часть рутинной работы

LINUX ИЗНУТРИ

Оборудование и Linux Новая страница: sysfs

Нам часто приходится слышать, что Linux «недостаточно хорошо поддерживает современное оборудование». **Владимир Попов** хочет привести пару подкрепленных примерами доводов в защиту открытой ОС.



Вам никогда не казалось парадоксальным, что Linux, который используется в мобильных телефонах и на суперкомпьютерах, пользователи IBM PC часто упрекают в том, что он «плохо поддерживает оборудование»? Не будем вспоминать, что для Microsoft (а ведь именно с Windows сравнивают Linux на персональных компьютерах) львиную долю работы по поддержке оборудования выполняют производители этого самого оборудования, просто попытаемся проанализировать, есть ли реальные основания для таких упрёков.

Настоящая статья представляет собой попытку «защиты» Linux, опирающуюся на информацию о новой драйверной модели, ставшей частью ядра версии 2.6

IBM PC: что внутри?

Для начала об оборудовании, непосредственно контактирующем с внутренними шинами IBM PC. Это те самые карты, которые раньше во множестве вставлялись непосредственно в слоты материнской платы. На настоящий момент большая часть этих устройств находится «на борту» – в составе микросхем, расположенных на материнской плате, но сути это не меняет. Каждое такое устройство использует определённые ресурсы – адреса памяти и портов ввода/вывода (о линиях прерывания и каналах прямого доступа к памяти (DMA) упоминать не будем, поскольку шина ISA усилиями Intel таки похоронена окончательно). Казалось бы: устройство скорее предоставляет ресурсы, чем использует их. Но поскольку адресное пространство у компьютера одно, то получается, что предоставляя ячейки памяти или регистры портов ввода/вывода, устройство использует именно адреса компьютера.

И хотя вышеупомянутых карт в составе IBM PC с каждым годом становилось всё меньше, количество устройств неизменно возрастало, и каждому требовались ресурсы. Совершенно очевидно, что нужен был механизм, который бы эти ресурсы распределял (хотя бы для того, чтобы обеспечить возможность работы в составе компьютера двух одинаковых устройств, требующих, естественно, одинаковых ресурсов). Такой механизм существует, и называется он – Plug-and-Play. Это может показаться странным, но такое несколько «игривое» название получило именно распределение ресурсов устройствам, а не подключение

устройства к компьютеру без перезагрузки последнего, как можно было бы предположить. Для такого «горячего» подключения используется совсем иной термин – hotplug. Первоначально о hotplug можно было говорить только в применении к USB и PCMCIA/Cardbus. Потом этот список был дополнен устройствами Firewire и CompactPCI, а в настоящее время hotplug возможен для «нормальных» PCI-устройств (посредством специальных hotplug-контроллеров), IDE (некоторые RAID контроллеры), SCSI, модулей памяти и, наконец, CPU. Вот уж, воистину: «всё смешалось в доме Облонских», как справедливо заметил великий русский писатель.

Основной шиной, посредством которой обмениваются информацией все устройства современного IBM PC, является PCI. Пользователи Linux могут легко в этом убедиться, набрав в командной строке **lspci**, **lspci -vv** или **scanpci -v**. Остальные шины, будь-то AGP или USB, получают доступ к PCI посредством контроллеров, называемых мостами. События, происходящие на этих «прочих» шинах, в принципе, не интересуют систему управления Plug-and-Play, поскольку не влекут за собой перераспределения шинных ресурсов. Так бы и сосуществовали мирно разные устройства: Plug-and-Play контролировал бы распределение внутренних ресурсов вычислительной системы, hotplug реагировал бы на подключения к шинам USB и PCMCIA, пока «горячее» подключение не стали использовать для PCI-устройств.

Кроме того, есть ещё одно обстоятельство, подталкивающее разработчиков Linux к пересмотру принципов работы с оборудованием. Дело в том, что, с точки зрения ОС, принципиально различным образом подключенные устройства, в общем-то, очень похожи. И тем, и другим требуется драйвер, да и представлены они, в конечном счёте, одинаково. Однако об этом – ниже.

Устройства Linux

Linux, как известно, унаследовал от UNIX представление устройств в виде файлов, находящихся в каталоге **/dev** и характеризующихся уникальным идентификаторами – числами **minor** и **major**. Существование таких файлов – абсолютно необходимое условие доступности устройств, и не так давно общепринятой практикой было их «заблаговременное» создание. В настоящий момент каталог **/dev**, заполненный подобным образом, выглядит просто необъятным, что, по меньшей мере, раздражает.

Кроме того, поскольку под номера **minor** и **major** изначально было выделено всего по байту, и в скором времени их перестанет хватать. Одним словом, изменения назрели, и, забегая вперёд, скажем, что **udev**, опирающийся на данные новой драйверной модели, доступные через *sysfs*, является очередной (после *devfs*) попыткой решения проблемы неуправляемого роста **/dev**.

Ещё одно условие использования устройства – это существование и загрузка соответствующего модуля ядра – драйвера. Модуль этот может быть включён в состав ядра во время компиляции последнего, но сути это не меняет: нет модуля – нет работоспособного устройства. Вот теперь картина получается вполне законченной:

- система Plug-and-Play (а в случае Linux это BIOS) обнаруживает устройство, конфигурирует его и выделяет ему необходимые ресурсы с учётом «пожеланий» всех остальных устройств;
- драйвер находит «своё» устройство, читает его конфигурацию, и после этого готов выполнять свои коммуникативные функции;
- драйвер должен быть «ассоциирован» с файлом устройства (а последний, в свою очередь, должен существовать), и только после этого обращение к файлу в **/dev**, фактически, станет обращением к устройству.

Для виртуальных устройств, каковыми являются сетевые адаптеры *ethN*, достаточно последовательно загружать командой **modprobe module_name** модули из каталога **/lib/modules/version/kernel/drivers/net**, пока вывод команды **cat /proc/net/dev** не укажет на то, что устройство **eth0** появилось. Именно так некоторые live-cd и поступают.

Для устройств, нуждающихся в наличии «своего» файла в **/dev**, потребуется наличие *devfs* и соответствующего демона *devfsd*, реагирующего на подключение устройства загрузкой соответствующего драйвера и созданием файла в **/dev**. Так, для звукового PCI-адаптера модули нужно брать из **/lib/modules/version/kernel/sound/pci**, а результат оценивать по по наличию устройства **/dev/dsp**.

Разумеется, существовали и более интеллектуальные способы поиска нужного драйвера для устройства, основанные на сопоставлении идентификаторов оборудования (**cat /proc/bus/pci/devices**) и наличия аналогичных данных модулей из файла **/lib/modules/version/kernel/modules.pcimap**, только то, что подходит для сетевого адаптера или звуковой карты, трудно приложить к неизвестному оборудованию: драйвер, положим, нашли и загрузили, а какой файл устройства создавать? Да и принципы работы *devfs* оказались не без изъяна. Чего стоило только «размножение» устройств при многократном подключении одного и того же USB-накопителя...

Новая драйверная модель Linux

Довольно серьёзные проблемы в работе с PnP устройствами преодолены в Linux к середине 2004-го года. Из не-PnP системы Linux превратился в систему с децентрализованным механизмом PnP: **Plug-and-Play** реализуется для каждого отдельного устройства. Теперь драйвер, обнаружив «своё» устройство, запрашивает закреплённые за ним ресурсы и, как правило, «соглашается» с таким распределением. Драйвер, используя функции ядра, может попытаться переопределить ресурсы устройства. В этом случае за отсутствие возможных конфликтов отвечает ядро – оно просто не принимает адреса, конфликтующие с другими устройствами.

Драйверная модель требовала изменений. Прежде всего, устранению подлежали следующие недостатки ядра 2.4:

- отсутствие единого метода представления связей между драйверами (модулями ядра) и устройствами;
- отсутствие общего hotplug-механизма, единого для PnP-конфигурируемых устройств и устройств, подключаемых к независимым шинам, таким как USB или PCMCIA;
- загромождение **procfs** информацией, не имеющей отношения к процессам. Фактически такой, для хранения которой она изначально не предназначалась.

Вообще-то, говорить следовало бы прежде всего об изменениях в архитектуре ядра и появлении новых объектов подсистемы обслуживания устройств (**kobjects**), но это всё «дела ядерные». Для большинства же пользователей наиболее заметным следствием внедрения новой драйверной модели стало появление *sysfs* – виртуальной файловой системы, экспортирующей в пространство пользователя информацию ядра о присутствующих в системе устройствах и драйверах. Автором *sysfs* (ранее именовавшейся *driverfs*) является Патрик Мошель (Patrick Mochel), а для больших систем ее дорабатывал Маниш Сони (Maneesh Sony).

Если вернуться к вышеперечисленным недостаткам ядра 2.4, то можно сказать, что первые два устраняются благодаря тому, что инфор- >>

UDEV

Основной автор и популяризатор *udev* – Грег Кроах-Хартман (Greg Kroah-Hartman). По его замыслу *udev* должен:

- выполняться в пространстве пользователя;
- динамически формировать **/dev**;
- генерировать произвольные и непротиворечивые имена устройств;
- обеспечить пользователя API доступа к информации об устройствах системы.

Едва ли не самым замечательным было то, что первая реализация *udev* представляла собой всего 6Kb откомпилированного кода: можно понять интерес производителей гаджетов...

Как это ни странно, но управление статическими узлами («нодами») устройств в **/dev**, основанное на информации ядра – лишь «сопутствующая выгода» использования *udev*. Разрабатывался-то он прежде всего для того, чтобы избавиться от ответственности за назначение **major/minor** номеров и получить возможность произвольного именования устройств. Первое достигается тем, что *udev*, в отличие от *devfsd* только «наследует» **major/minor**, отвечает же за них исключительно ядро. А механизм управления именованием получился настолько удачным, что подобным образом, в конце концов, стали создавать символические ссылки и запускать скрипты. В конечном счёте, оказалось, что такой «вооружённый» скриптами *udev* вполне может обойтись без **/sbin/hotplug** и такой подход становится в последнее время доминирующим. По крайней мере, он уже принят на вооружение Mandriva, Ubuntu, Arch Linux...

<< мация как об устройствах, так и о загруженных драйверах, представляется в рамках одной файловой системы. Причём в список устройств включаются устройства на всех шинах IBM PC. *procfs* же разгружается потому, что *sysfs* монтируется не как подкаталог **/proc**, а как каталог **/sys**.

sysfs не может быть модулем, она всегда – часть ядра. Для обеспечения доступа к ней нужно выполнить команду **mount -t sysfs sysfs /sys**. Обычно это монтирование выполняется первым же стартовым скриптом.

Существует непосредственная связь между *sysfs* и объектами инфраструктуры ядра. Всем зарегистрированным системой объектам инфраструктуры (тем самым **kobjects**) в **/sys** соответствуют свои каталоги. Форма дерева каталогов, берущего своё начало в **/sys**, максимально точно соответствует соотношению между объектами. Активно используемые символические ссылки делают это дерево несколько «развесистым», но зато отражают отношения наследования и связи между объектами в максимальной степени.

Если каждое устройство или драйвер – каталог, то файлы каталогов – атрибуты этих объектов. Таким образом, *sysfs* превращает операции ввода/вывода для этих файлов в методы атрибутов объектов ядра. Операции записи/чтения атрибутов **kobjects** становятся такими же доступными, как операции записи и чтения файлов.

Файлы атрибутов – текстовые. Существует правило, в соответствии с которым одному атрибуту соответствует один файл. Поскольку хранение по одному значению в файле может оказаться неэффективным, то «на будущее» приветствуется выражение атрибутов в форме массива значений одного типа. Нужно сказать, что **sysfs** родилась не на пустом месте, и если информация о связях объектов ядра – это, действительно, новость, то большую часть атрибутов устройств можно было получить и в «старой» **/proc/sys**. Файлы атрибутов в **/sys**, однако, существенно отличаются от файлов **procfs**, синтаксический анализ содержимого которых мог быть довольно сложным.

Структура *sysfs* выражает соотношения структур данных ядра. Названия подкаталогов **/sys** говорят сами за себя:

- **devices/** – полностью соответствует внутреннему дереву устройств ядра, а символические ссылки в подкаталогах (когда они есть, разумеется) указывают на шину устройства, принадлежность его к определённому классу, соответствующий загруженный драйвер и т.п. Дерево может быть достаточно сложным и отражает связь между устройствами;
- **bus/** – представляет собой перечень шин, зарегистрированных в ядре. Каталог каждой шины содержит подкаталоги **devices/** и **drivers/**. Причём,
- **devices/** – это символические ссылки на каталоги устройств, описанных в системе (реально расположенных в **/sys/devices/...**);
- **drivers/** – каталоги драйверов, загруженных для устройств, подключенных к данной шине. Каждый такой каталог содержит, как минимум, пару файлов-атрибутов **bind** и **unbind**, предназначенных для управления драйвером (*см. LXF73*), а когда драйвер обнаруживает «своё» устройство, то в каталоге появляется символическая ссылка на каталог этого устройства.
- **block/** содержит каталоги всех блочных устройств, присутствующих в настоящее время в системе. В данном случае под устройством понимается совокупность физического устройства и драйвера. То есть, если при подключении USB-накопителя некоторое новое устройство в **/sys/devices/** появится всегда (можно говорить о наличии физического устройства), то появление каталога **/sys/block/sda** зависит ещё и от наличия в памяти необходимых драйверов (usb-storage, sd_mod и т.д. – включая все драйвера, необходимые для поддержки usb);
- **class/** – отражает группировку устройств в классы. Всякое подключенное устройство создаст новый подкаталог в дереве **/sys/class**. Как и в предыдущем случае, подразумевается наличие и устройства, и его драйвера;
- остальные каталоги могут быть очень полезны. Так, в **module/** можно узнать атрибуты, а в некоторых случаях даже изменить параметры загруженных модулей. Через файлы **power/** осуществляется перевод системы в режим **suspend**, но к определению оборудования содержимое остальных каталогов отношения уже не имеет.

Более детальную информацию об особенностях модели драйверов при желании можно найти в файлах каталога **/usr/src/linux/Documentation/driver-model**. Обширной, однако, её пока не назовёшь. Да и вообще, нельзя сказать, что информация в **/sys** предназначена для «простых пользователей». *Sysfs* используется системными утилитами, такими, как udev и HAL, обеспечивая их доступом к информации об устройствах и их драйверах. Модифицируя атрибуты устройств и драйверов, udev и HAL теперь могут даже изменять конфигурацию последних.

Что из этого следует?

Из того факта, что существование *sysfs*, в принципе, может быть проигнорировано «простым» пользователем, совершенно не следует, что новая драйверная модель не отражается на «повседневном» использовании Linux. Окончательный переход на ядро 2.6 влечёт за собой неминуемое изменение архитектуры дистрибутивов, предлагаемых их разработчиками. Внешние изменения в системе управления загрузкой модулей (новые расширения файлов модулей, **modprobe.conf** вместо **modules.conf**) отнюдь не случайны: они введены для обеспечения возможности временного сосуществования старой и новой моделей.

Вслед за вышеупомянутыми udev и HAL (ближайшими к драйверной модели ядра утилитами), изменения косаются определителей оборудования, инсталляторов, LiveCD – всех, кому приходится заниматься конфигурацией системы.

На настоящий момент загрузка драйверов для постоянно присутствующего в системе оборудования обеспечивается либо командами **«modprobe module_name»** в стартовых скриптах, либо строками **«alias string module_name»** в **modprobe.conf**, либо принудительным запуском *hotplug* (т.наз. coldplug, когда hotplug-события синтезируются в ходе загрузки системы), либо комбинацией всех трёх способов. За загрузку драйверов для подключаемых «на ходу» устройств, отвечает, обычно, *hotplug*. Тот же *hotplug* генерирует события для *udev*, заставляя последний создавать для вновь появившегося устройства файл в **/dev**. Всё это, разумеется, происходит при участии ядра. Интерфейсом же между ядром и утилитами является, как вы, наверное, догадываетесь, *sysfs*.

Новая драйверная модель и рост вычислительных возможностей IBM PC делают возможным упрощение этой схемы. Судите сами: что такое десяток-другой килобайт модуля драйвера для ЭВМ с гигабайтом памяти? Так почему бы не загрузить модули «с запасом»? **udev/HAL** останется только создавать в **/dev** соответствующий файл при подключении устройства, *hotplug* же, таким образом, становится вообще не нужен.

Ещё один вариант: возложить на *udev* обработку **hotplug**-событий. Почему бы не устранить промежуточное звено, тем более, что стиль управления работой *udev* представляется более гибким? Очевидно, всё определяется назначением целевой системы. Так, домашний компьютер стоит настроить «вручную», поскольку потеря времени на определение устройств при каждой загрузке раздражает («ну что ты там ищешь? ничего нового за ночь не выросло»), а LiveCD должен определять оборудование «с нуля».

ДРАЙВЕРА ОТ ПРОИЗВОДИТЕЛЕЙ ДЛЯ LINUX

Не нужно думать, что производители оборудования не делают драйвера для Linux «из вредности». Просто поставка драйвера в форме двоичного модуля для «константного» ядра XP и предоставление спецификации, достаточной для построения модуля для произвольного ядра Linux – совершенно разные вещи. Существует, однако, компромисс в форме драйвера, часть из которого предоставляется в двоичном виде (и «закрыта», таким образом, от потенциальных конкурентов), а часть – в виде исходных текстов. Причём, именно вторая часть обеспечивает возможность «подгонки» создаваемого модуля к любому ядру. В таком виде предоставляет драйвера NVIDIA, за что мы ей весьма благодарны. К сожалению, сказать, что её примеру последовали многие, нельзя.

Практикум

В качестве практикума предлагаю посредством анализа **/sys** рассмотреть программную поддержку **ide**-устройств. Наиболее полно дерево устройств представлено, как уже говорилось, в каталоге **/sys/devices**. Однако полнота представления может быть и утомительной. Поэтому, оставив в стороне каталоги **/sys/block** и **/sys/class** (их содержимое зависит как от наличия собственно устройств, так и от присутствия драйверов, а хотелось бы пройти весь путь исключительно от «железа»), сосредоточимся на **/sys/bus**.

Переходим в подкаталог **ide/**. Поскольку управлять контроллерами через *sysfs* мы не планируем, то интересовать нас будут исключительно каталоги. Набираем: **tree -d** (ключ -d отфильтровывает файлы) – и вот, что мы имеем:

```
.
|-- devices
|   |-- 0.0 -> ../../../devices/pci0000:00/0000:00:1f.1/ide0/0.0
|   `-- 0.1 -> ../../../devices/pci0000:00/0000:00:1f.1/ide0/0.1
`-- drivers
    |-- ide-cdrom
    |-- ide-disk
    |   `-- 0.0 -> ../../../../devices/pci0000:00/0000:00:1f.1/ide0/0.0
    `-- ide-scsi
        `-- 0.1 -> ../../../../devices/pci0000:00/0000:00:1f.1/ide0/0.1
```

«Стрелочки», как вы, вероятно, догадались, обозначают символические ссылки. Анализ прост до примитивности: имеем два **ide**-устройства и три драйвера. О том, что одно из устройств – диск, а второе – CD-ROM, также говорить излишне. То, что драйвер *ide-disk* обслуживает винчестер – естественно, а вот то, что CD-ROM обслуживается драйвером *ide-scsi*, уже представляет некоторый интерес. И не потому, что CD-ROM на самом деле подключен к шине IDE (*cdrecord* рекомендует эмулировать *ide-cdrom* как scsi-устройство), а потому, что, если память не изменяет, ядро 2.6 советует от этой эмуляции уже отказаться. Отметим это обстоятельство (нужно будет попробовать) и отправимся по ссылке **../../../devices/pci0000:00/0000:00:1f.1/ide0/0.0**, поскольку в **/sys/bus/ide** ничего интересного больше нет. «По пути» к этому каталогу советую внимательно «смотреть по сторонам». Так мы узнаем, что контроллер ide «сидит» на шине PCI (о чём и так можно было догадаться) и имеет идентификатор 0000:00:1f.1. Способ формирования идентификатора особого интереса не представляет, да и потребуется он только в том случае, если мы захотим сопоставить наблюдаемые данные с результатом вывода упоминавшейся выше *lspci*. В каталоге **/sys/devices/pci0000:00/0000:00:1f.1/** исследовав символические ссылки, можно узнать, что драйвер, используемый PCI-контроллером, – PIIX_IDE. Спустившись в **ide0**, увидим, что контроллер этот обслуживает два устройства, младшее из которых (0.0) и является винчестером. Символические ссылки указывают на то, что устройство идентифицируется в системе как **/dev/hda**, принадлежит шине ide и обслуживается драйвером *ide-disk*. Последовав по символической ссылке **block/**, можно получить данные о разметке диска.

Аналогичный поход для CD-ROM приведёт нас к устройству **ide0/0.1**, которое окажется scsi (а что ещё можно было ожидать от драйвера *ide-scsi*?). В соответствии с архитектурой scsi, мы обнаружим **host0** (принадлежащий классу scsi_host, разумеется). **host0**, в свою очередь, содержит **target0:0:0**, единственное устройство которого (0:0:0:0) и является нашим CD-ROM. Он принадлежит шине scsi, идентифицируется в системе как **/dev/sr0**, обслуживается драйвером **sr** и входит в класс scsi_generic с именем **sg0**. Перебрав атрибуты, можно найти и модель, и тип, и производителя привода. Пожалуй, этого достаточно. Прошу прощения, если утомил. Зато, надеюсь, убедил в том, что *sysfs* предоставляет предостаточно информации для того, чтобы идентифицировать устройство, определить модули, которые обеспечивают его программную поддержку и определить положение в дереве устройств.

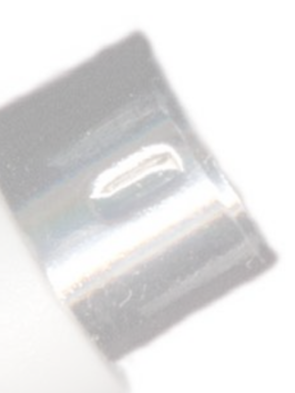
И ещё один пример для самых любознательных. На сей раз – более полезный.

Предлагаю попытаться подключить, скажем, USB-IrDA Bridge неизвестного производителя из одной большой восточной страны. USB-IrDA Bridge – это такая штуковина, которая, будучи подключенной к USB-разъёму, эмулирует ИК-порт. Для связи с мобильным телефоном, например. Ни о каком драйвере производителя и речи быть не может, да и опознавательных знаков всего два: «IrDA» и «Made in China». Итак...

Начать, как и в предыдущем случае, предлагаю с **/sys/bus**. Интересующий каталог, разумеется – **usb/**. Поскольку драйвер для моего «подопытного» в системе заведомо отсутствует, то переходим сразу в каталог **devices/**. USB-контроллеры, коих у меня обнаружилось целых пять, представляют из себя эдакие «дуальные» устройства: с одной стороны это мосто между шинами PCI и USB, а с другой – первое устройство на USB-шине, называемое, обычно, корневым концентратором (root hub). Оба «лица» представлены в каталоге **/sys/bus/usb/devices/** в виде символических ссылок. Мосты имеют «собственные» имена (**usb1**..**usb5**), а все hub-ы (как, впрочем и остальные устройства, подключаемые к usb-шине) представлены численными идентификаторами вида **N-P:C.I**, где **N** – номер моста (он же – номер hub-а), **P** – номер порта hub-а, вообще-то соответствующий определённому usb-разъёму, а для самого hub-а всегда равный нулю. **C** – номер конфигурации и **I** – номер интерфейса. Мосты обнаруживаются драйвером *usb*, hub-ы – драйвером *hub*.

Любое подключение usb-устройства приводит к появлению в каталоге новых ссылок. Так, в частности, подключение «подопытного» устройства привело к появлению ссылки **2-1:1.0**, что означает: к первому порту второго hub-а подключено устройство, конфигурация – 1, интерфейс – 0. Переход в каталог этого устройства подтвердит, что нужный драйвер не загружен – ссылки **driver** не существует. Теперь попытаемся применить подход, описанный для сетевых карт: командой modprobe последовательно загружаем модули из каталога **/lib/modules/2.6.15/kernel/drivers/net/irda**, благо их там пока всего 12. Если повезёт, то после загрузки очередного модуля (в моём случае это оказался stir4200) ссылка **driver** появится. Более того, в моём случае появилась ссылка **net:irda0**, что говорит о том, что в системе обнаружено устройство класса net с именем **irda0**. С чем себя и поздравляю.

Это, правда, не означает, что вы тут же сможете подключить и мобильный телефон: пока это только основание для уверенности, что связь по ИК-каналу между телефоном и вашей Linux-системой возможна. Однако, это уже совсем другая история...



Выводы

Драйверная модель ядра Linux версии 2.6.x предоставляет, без сомнения, новые возможности для определения оборудования. В некоторых случаях можно говорить о новом подходе к написанию инсталляторов и LiveCD. Можно также рассчитывать на увеличение количества модулей поддержки оборудования в составе ядра, поскольку одной из целей внедрения модели было упрощение разработки драйверов.

С другой стороны, мне кажется, что в данном случае, вслед за разработчиками ядра, приверженцы Linux выбирают «не Windows-way». С новыми возможностями ядра специалист получит дополнительные возможности для оптимизации системы, а «простой пользователь» опять станет заложником создателей дистрибутивов – как скоро (и насколько успешно) они интегрируют эти самые возможности.

Иными словами, Linux продолжает развиваться в направлении предоставления больших возможностей для создания систем различного назначения и сложности, а не как система, ориентированная на конечного пользователя офисного или домашнего компьютера. Последнего, будем надеяться, порадуют своими разработками составители «пользователь-ориентированных» дистрибутивов – ведь расширение возможностей системы означает и большие возможности для создания идеального настольного компьютера. Но не будем требовать от них слишком многого: динамично развивающееся ядро в который раз заставляет пересматривать архитектуру системы. LXF

PNP ОС – А ХОРОШО БЫ...

Есть, однако, функции PnP ОС (ОС с централизованной системой Plag-n-Play), которые хотелось бы увидеть в Linux:

- распределение ресурсов при их дефиците (утрачивает актуальность с каждым годом);
- выбор драйвера при наличии альтернативы для одного и того же устройства;
- поиск драйвера вместо ожидания реакции программы, ответственной за его загрузку;
- дальнейшее облегчение работы разработчиков драйверов.

Перспектива роста – налицо. Подождём...

ИСПОЛЬЗОВАНИЕ NDIS-ДРАЙВЕРОВ

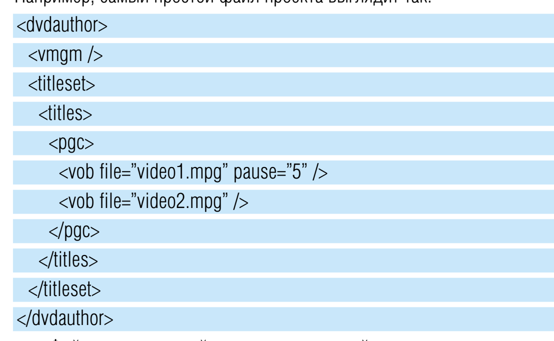
Покупая сетевую карту вы, как правило, оплачиваете и прилагаемый драйвер. Вот только драйвер этот – ndis, т.е. для MS Windows. Иногда, однако, и эти драйвера удаётся успешно использовать в Linux. Лучшие примеры – это ***DriverLoader*** (http://www.linuxant.com/driverloader/) и ***ndiswrapper*** (http://ndiswrapper.sourceforge.net/), успешно применяемые для множества Wi-Fi адаптеров. Это – второй известный мне случай использования в Linux проприетарного ПО, написанного для MS Windows. Первый – видео кодеки в MPlayer.

МУЛЬТИМЕДИА-ТЕХНОЛОГИИ

Создание DVD в Linux

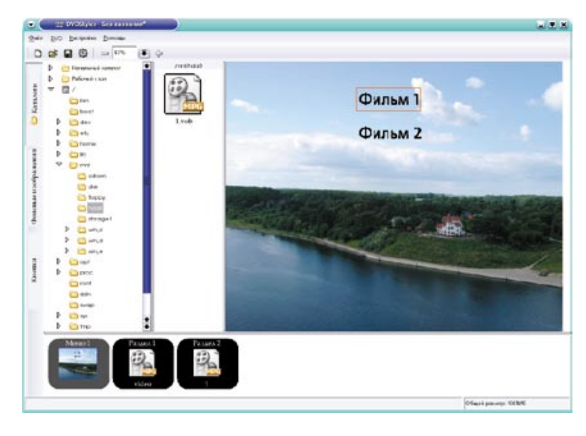
Долгое время создание DVD-дисков в Linux было далеко нетривиальной задачей, отнимающей много времени. Теперь ситуация изменилась в лучшую сторону, появились инструменты, позволяющие быстро и удобно создавать авторские DVD.С одним из них вас познакомит **Сергей Яремчук**.

Закончились праздники, оцифровано, отредактировано и перекодировано не один час видео. Что теперь? Самое простое решение – записать все это на диск как обычные данные. Но не все проигрыватели способны понять такой формат, кроме того, хочется сделать свое творение более презентабельным и приятным для просмотра. А значит, нужны меню, внешнее оформление и прочие мелочи, делающие просмотр комфортным. Решено: будем создавать DVD! Основным инструментом предназначенным для создания DVD-дисков, является *DVD Authoring Tools* (dvdauthor), активно развивающийся с июля 2002 года проект, основным разработчиком которого является Скотт Смит (Scott Smith). Сайт dvdauthor расположен по адресу: **http://dvdauthor.sourceforge.net/**. Остальные встречающиеся в Интернете приложения, как правило, являются всего лишь оболочками или вспомогательными утилитами, облегчающими работу с dvdauthor. Вот только пользоваться им при создании дисков – далеко не простая задача. Пользователю необходимо самому создать готовый управляющий XML-файл проекта, необходимый для генерации содержимого, подкаталоги и еще кучу мелочей, требующих внимания. Например, самый простой файл проекта выглядит так.

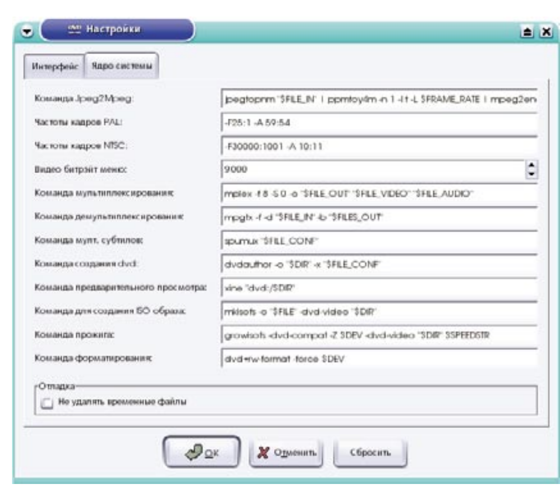


```
<dvdauthor>
  <vmgm />
  <titleset>
    <titles>
      <pgc>
        <vob file="video1.mpg" pause="5" />
        <vob file="video2.mpg" />
      </pgc>
    </titles>
  </titleset>
</dvdauthor>
```

Файл, описывающий проект, содержащий несколько разветвленных меню, фоновую музыку и картинки может занимать несколько



В настройках можно подсмотреть и подправить параметры.



Основное окно *DVDStyler*.

килобайт. И сами понимаете, что ручное его составление займет приличное количество времени. Утилиты вроде **dvd-slideshow** (см. «*Полезные ссылки*») упрощают создание меню и позволяют создать видеофайл из графических изображений.

Графические интерфейсы к DVDauthor

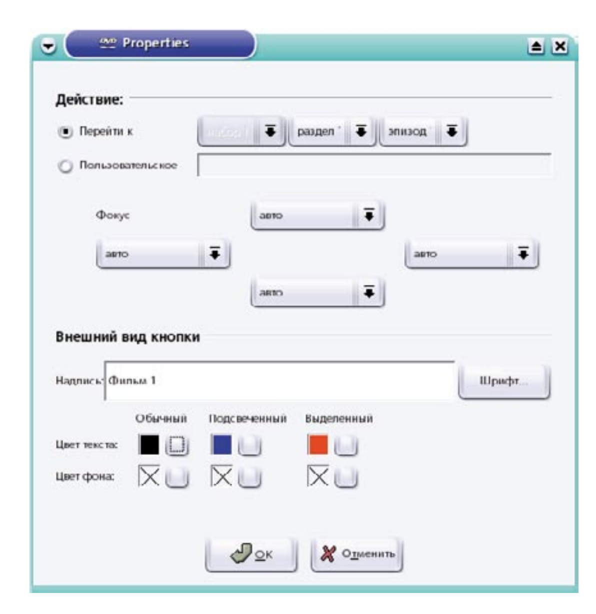
Существует несколько проектов, реализующих графический интерфейс к программам для создания DVD. Например, *Q-DVD-Author* на базе *Qt*, *DVDStyler*, написанный с использованием wxWidgets, Java-приложение Varsha [5] и, наконец, *Polidori*, базирующийся на GTK2. Все они являются относительно молодыми проектами, находящимися на раннем бета (а то и альфа) этапе своего развития и обладающие различными возможностями и стабильностью в работе. Среди них наиболее функциональным является *Q-DVD-Author*, последняя бета-версия которого, 0.1.0, вышла в декабре 2005 года и работает уже вполне стабильно, а потому может быть рекомендована к применению. Проект *Polidori* наглухо застрял в стадии «альфа» и уже два года как не развивается. У автора после успешной компиляции он наотрез отказался запускаться, не поясняя причин, хотя судя по описаниям, его возможности также неплохи. Свое первоначальное ознакомление с созданием DVD советую начать с *DVDStyler* Алекса Туринга (Alex Thuering). Хотя возможности последнего не столь обширны, как у некоторых из названных проектов, их наверняка хватит для создания большинства любительских DVD. Зато его интерфейс мне кажется наиболее понятным и логичным, вероятно, все дело здесь в простоте. Поддержка drag and drop для MPEG-видеофайлов (к сожалению, *DVDStyler*, в отличие от того же Q-DVD-Author, плохо умеет автоматически определять и перекодировать видео из других форматов), импорт изображений для фона, создание NTSC/PAL меню с использованием любых доступных шрифтов и цветов, добавление post-команды которая будет выполнена

по окончании воспроизведения любого видео – вот и весь ассортимент возможностей. Последней на момент написания статьи версией была 1.4, что само по себе говорит о стабильности продукта. Кроме Unix-систем, доступна версия и под Windows (естественно, бесплатная). Работа *DVDStyler* практически незаметна даже на маломощных компьютерах. С остальными мой компьютер с процессором 1ГГц явно «подтормаживает».

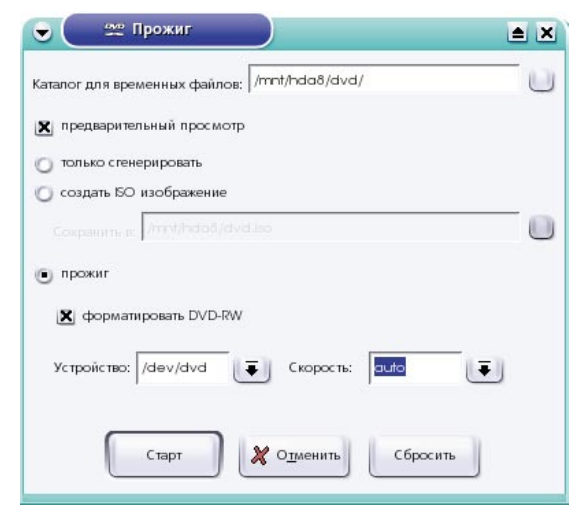
Запускаем DVDStyler

Для установки *DVDStyler* кроме него самого и библиотек интерфейса wxGTK потребуются инструменты для кодирования mpeg-потоков и записи DVD-дисков: mpgtx, cdrtools, dvd+rw-tools и Mjpeg Tools. К сожалению, в используемом мной при написании статьи дистрибутиве *ALTLinux Master*, ярлык для запуска не встраивается в меню KDE. Как это будет с Gnome или в других дистрибутивах, мне неизвестно поэтому мы просто наберем в окне терминала **dvdstyler**. При первом запуске, программа спросит на каком языке с вами общаться. Русский в этом списке есть, выбираем **тХУУЛЙК [ru]** (да, именно так) и нажимаем OK. Все теперь, программа будет использовать локализованный интерфейс. Если вместо читаемого текста будут непонятные знаки, значит, требуется изменить системную локаль. Для выхода из сложившейся ситуации можно использовать конфигурационный файл, который находится в домашнем каталоге пользователя. Называется он **.dvdstyler** (а вы что ожидали?). Хранящийся в нем параметр **Language=56** соответствует английскому языку, поставив 156, мы изменим его на русский, после чего следует перезапустить приложение. Можно также использовать "Настройка – Параметры" (Configuration – Setting). Здесь во вкладке "Интерфейс" настраивается язык, формат меню (PAL или NTSC) и указывается каталог для хранения временных файлов.

Обратите внимание, что во вкладке "Ядро системы" доступны все команды, которыми вы будете оперировать при помощи пунктов меню (*рис.1*). Здесь вы можете не только познакомиться с ними, но и отредактировать под свои нужды. И, наконец, главное окно программы (*рис.2*). Окно визуально поделено на три основных части. В центре располагается главное окно редактирования, где можно предварительно оценить результат. Внизу показан список используемых объектов с их уменьшенными копиями. Слева находится браузер, при помощи которого выбираются файлы. Браузер имеет три вкладки: Каталоги, Фоновые изображения и Кнопки. При помощи двух последних можно быстро осуществить доступ к заранее подготовленным объектам, которые хранятся в **/usr/local/share/dvdstyler** (если не указан другой путь при настройке программы). Для быстрого доступа рекомендуется положить туда все, что требуется (это должны быть jpeg-файлы размером 720x576), чтобы не лазить каждый раз по всему диску в поисках фоновых изображений.



Кнопки бывают разные.



Параметры создания DVD.

Создаем диск

Для создания нового проекта выбираем "Файл-Новый" и указываем видеоформат проекта. Затем при помощи проводника перетаскиваем в нижнюю панель проекта файлы с фильмами. Для создания меню выбираем "Фоновые изображения" и перетаскиваем в главное окно понравившееся изображение и вручную центрируем его. Теперь переходим ко вкладке "Кнопки" и переносим модель кнопки в главное окно, а затем помещаем ее на свое место. Щелчком левой кнопки по ее изображению вызываем контекстное меню и выбираем свойства объекта (*рис.3*). Заполняем их. Так как на форме у нас помещено два фильма, то указываем переход в "Раздел 1" и "Эпизод 1", то есть в начало первого фильма. Указываем свойства шрифта и название кнопки. Подтверждаем изменение нажатием OK. Таким же образом создаем вторую кнопку, осуществляющую переход ко второму фильму, выставив соответствующим образом ее свойства. К сожалению, *DVDStyler* в отличие от того же *Q-DVD-Author* пока не умеет использовать в кнопках меню картинки и музыку, но эта возможность планируется в будущем. Кстати, кнопки меню не обязательно должны вызывать фильмы. При большом количестве видеоматериала можно использовать разветвленную систему подменю. При необходимости можно задать другие свойства проекта, например автоматический переход к первому фильму при загрузке, свойства аудио и видео. Для этого необходимо щелкнуть в главном окне левой кнопкой мыши. Теперь, когда проект готов сохраняем его (Файл-Сохранить) на диске. Кстати, полученный файл можно использовать и индивидуально, генерируя необходимую структуру непосредственно с **dvdauthor**. Например.

```
$ dvdauthor –x dvd.xml –o /home/sergej/temp
```

И, наконец, создаем DVD, выбрав "Файл – Прожиг DVD", после чего появится меню, в котором нужно указать параметры (*рис.4*). Выставив галочку напротив пункта "Предварительный просмотр" вы получите возможность предварительно ознакомиться с результатом своей работы. По умолчанию для предпросмотра используется видеопроигрыватель XINE, запускаемый командой **xine dvd:/**, при желании, это можно изменить, покопавшись в настройках. Как видно из рисунка, можно просто сгенерировать проект, создав необходимую структуру каталогов, можно скомпоновать ISO-образ и записать его на DVD-диск. После нажатия кнопки OK будет автоматически очищен временный каталог, поэтому, если в нем был старый проект и он вам еще нужен, сохраните его в другое место. Затем начнется собственно создание DVD-диска. Эта операция может занять некоторое время, особенно на слабых компьютерах. После предпроосмотра результата создается ISO-образ и производится запись на диск. В любой момент процесс можно остановить и, подправив, начать все сначала. Все! Ваш первый DVD в Linux создан. Успехов! LXF

ПОЛЕЗНЫЕ ССЫЛКИ

- DVD Authoring Tools (dvdauthor) **http://dvdauthor.sourceforge.net/**
- dvd-slideshow **http://dvd-slideshow.sourceforge.net/**
- Q-DVD-Author **http://qdvdauthor.sourceforge.net/**
- DVDStyler **http://dvdstyler.sourceforge.net/**
- Varsha **http://sourceforge.net/projects/varsha**
- Polidori **http://polidori.sourceforge.net/**
- wxGTK **http://www.wxwidgets.org/**
- mpgtx **http://mpgtx.sourceforge.net/**
- cdrtools **ftp://ftp.berlios.de/pub/cdrecord/**
- dvd+rw-tools **http://fy.chalmers.se/~appro/linux/ DVD+RW/)**
- Mjpeg Tools **http://mjpeg.sourceforge.net/**

Вопрос? Ответ!

Если вы завязли в какой-то проблеме и чтение HOWTO не помогает, почему бы не написать нам? Наши эксперты помогут разобраться даже в самых сложных проблемах.

НАШИ ЭКСПЕРТЫ

Наши эксперты найдут ответ на самый трудный ваш вопрос. Если у вас проблемы с установкой, настройкой модема, сетью или еще чем-нибудь – просто напишите нам, а обо всем остальном позаботимся мы.



Управляя Интернет-провайдером, а заодно подрабатывая редактором дисков *LXF*, **Нейл Ботвик** (Neil Bothwick) скромно зовет себя мастером на все руки.

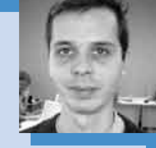


Брэндон Калигари (Brendon Caligari) больше десяти лет работает с Linux, он администратор работающей системы в фирме Rackspace Managed Hosting.



Сергей Владимирович Рыжиков

Генеральный директор компании «БИтрикс»



Максим Владимирович Смирнов

Ведущий программист отдела разработки компании «БИтрикс»



Александр К. – сторонник Unix-way. Молодой, но перспективный член дружной команды экспертов.



Валентин Синицын Поддерживает проект Slackware Reiser4, интересуется настольными Linux-технологиями и рад помочь Вам разобраться с ними.

Лучший форвард

В У меня четыре компьютера, один из которых работает как шлюз. Он оборудован двумя сетевыми картами, и на нем установлен Mandriva Linux 2005. Соединение с провайдером происходит через интерфейс eth0 и постоянный IP-адрес, а через eth1 я раздаю доступ в сеть на другие три компьютера, используя соответствующую утилиту в *Mandriva Control Center*.

Я хотел бы, чтобы на клиентских машинах были доступны некоторые сервисы, которые требуют настройки форвардинга: например тот же SSH.

Как это сделать? Есть ли какое-нибудь пошаговое руководство для настройки форвардинга через простую утилиту в Mandriva?

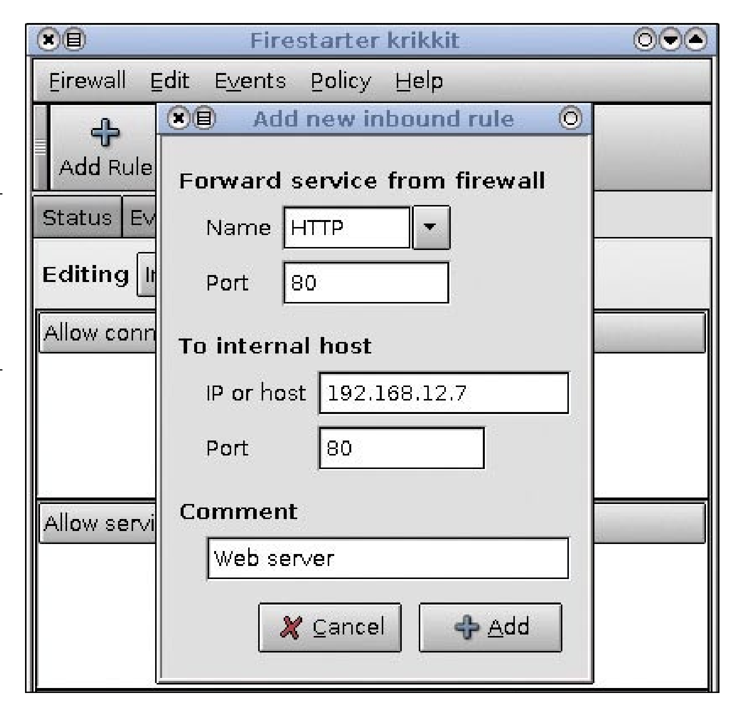
Джодо, с форума LXF

О Утилита из *Центра управления* Mandriva не имеет продвинутых возможностей для настройки полноценного шлюза. Но альтернативы есть: пожалуй, лучше всего Вам подойдет *Firestarter*. Эта программа поможет настроить шлюз, сконфигурировать роутинг и брандмауэр, и все это делается в простом GUI.

Firestarter находится в репозитарии **contrib**. Если этого репозитария нет в Вашем *Центре Управления Mandriva*, зайдите на **http://easyurpmi.zarb.org** и ознакомьтесь с инструкциями по его добавлению (добавьте заодно и репозитарий **PLF**). Теперь установите *Firestarter* из *Центра Управления Mandriva* – и стартуйте. Если в меню нету пункта *Firestarter*, запустите его из root-терминала.

Включившийся мастер загрузки предложит открыть доступ в Интернет для вашей сети; перед тем, как это сделать, отключите «раздачу» в *Центре Управления Mandriva*. Перейдите на вкладку правил (Policy), щелкните на нижней секции окна и выберите Add Rule (Добавить правило). Для стандартных сервисов вроде SSH нужно просто выбрать его в ниспадающем списке и указать IP-адрес.

Наконец, зайдите в *Центр Управления Mandriva* -> Система (System) -> Сервисы (Services) и убедитесь, что *Firestarter* автоматически запускается при загрузке системы, тогда форвардинг будет доступен.



***Центр управления* Mandriva справляется со шлюзом, но возможности *Firestarter* гораздо шире.**

Рекомендуем изучить документацию к *Firestarter*, чтобы подробнее узнать о других опциях. *Firestarter* можно использовать и как брандмауэр, благо он гораздо более гибок, чем брандмауэр *Mandriva Control Center*. ***НБ***

Беды Красной Шапочки

В Я хочу настроить двойную загрузку, с XP и Red Hat Linux, и столкнулся с такой проблемой: RH9 не опознает мой SATA-диск. Где найти для него драйвер, и как его загрузить, если в CD-ROM-приводе находится компакт-диск RH, а дисковода у меня нет? У меня материнская плата MSI Neo2 Platinum с процессором AMD64 3000+ (socket 939), видеокарта Nvidia 6800GT-AGP и 2 ГБ ОЗУ.

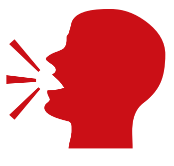
На другой машине с IDE-диском я загружал *Quake 3* на RH9, но когда я попытался установить *Quake 3* на Fedora Core 4, произошла ошибка. Друг рассказал мне, что дело в каком-то *Glibc*. Можете ли вы помочь?

Рой, Малайзия

О Red Hat 9.0 вышел несколько лет назад, еще до Вашей материнской платы. Потому установщик и не определяет Ваш жесткий диск. Вы уверены, что правильно выбрали дистрибутив? Вам явно не нужен сервер, с вашей-то видеокартой.

Я рекомендую Вам подобрать более современный дистрибутив, подходящий для настольной системы и имеющий 64-разрядную версию. Что-нибудь вроде SUSE 10, Mandriva 2006 или Ubuntu. Вы можете ознакомиться с полным списком на сайте **www.distrowatch.com**.

К сожалению, на второй вопрос вряд ли можно ответить, не зная даже, что за сообщение об ошибке выдал инсталлятор. Если Вы пришлете нам это сообщение, мы, скорее всего, сумеем разобраться. Это относится ко всем подобным письмам – сообщайте побольше информации, помогите нам Вам помочь. ***НБ***



Драйвер вслепую

В Я только что обновил мою старую видеокарту, установив вместо нее NVIDIA GeForce MX440 (128 МБ). Под Windows после установки драйвера все заработало хорошо, и теперь мне нужно установить Linux-драйвер, скачанный в раздел **/home**, на SUSE 9.3. Проблема в том, что мне надо установить их из командной строки, без запущенного X-сервера. Как это сделать?

Карл Дрейпер

О Нажмите Ctrl+Alt+F1 (для переключения в виртуальную консоль), войдите в систему от имени суперпользователя и выполните следующие команды:

```
init 3
cd ~carl #короче, ваше имя пользователя
sh NVIDIA-Linux<tab>
```

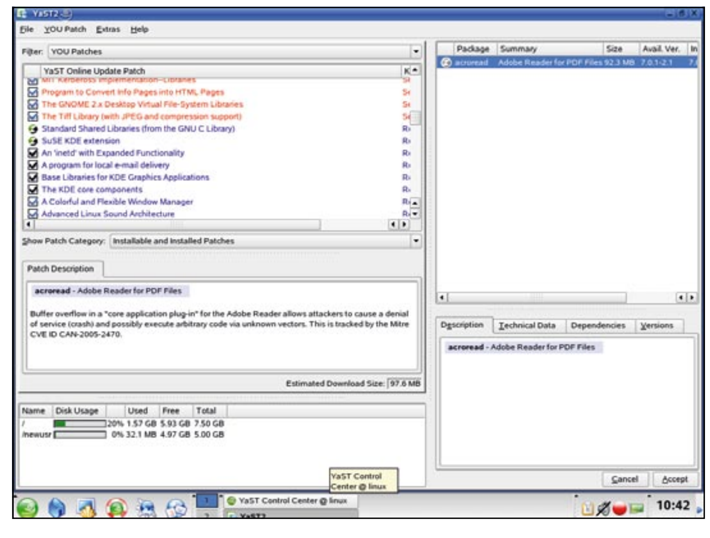
Клавиша табуляции служит для автодополнения пути. Ответьте на вопросы (сгодятся ответы по умолчанию). После этого нужно будет отредактировать файл **/etc/X11/xorg.conf** (под root), чтобы X-сервер использовал новые драйверы. Что именно нужно редактировать, лучше посмотреть в Readme-файле, поставляемом с драйвером. Перед редактированием на всякий случай создайте резервную копию **xorg.conf**, если драйвер по какой-либо причине откажется работать.

Если Вам некомфортно в консольном редакторе (хотя, если уж иметь дело с Linux, не худо его и освоить), выполните **init 5** для запуска графической системы. После этого выберите в главном меню System (Система)-> File Manager (Файловый менеджер) -> File Manager Super User mode. Перейдите к **/etc/X11**, отыщите **xorg.conf** и отредактируйте его с помощью *Kedit*. Сохраните файл и выполните **init 3** для выхода из графического режима.

Какой бы метод Вы ни выбрали для редактирования, запускайте X-сервер с помощью **init 5**. Если драйвер заработает, перед запуском X-сервера появится логотип NVIDIA.

Вместо установки вручную можно установить драйвер через *Yast*, следуя инструкциям на **www.suse.de/~sndirsch/nvidia-installer-HOWTO.html#5**. Но имейте в виду, что тогда версия драйвера будет не самой новой. ***НБ***

Исходный код драйверов Nvidia закрыт, и они не включаются в большинство дистрибутивов.



В SUSE можно установить что угодно при помощи *Yast*, но самые свежие драйверы Nvidia для видеокарт – на www.nvidia.com.



Время менять BIOS

В Уже три раза пытался установить Linux. Я купил CD/DVD-редакцию SUSE из-за рекламной кампании Novell. Установка шла как полагается до первой загрузки с диска. Пробовал переустанавливать SUSE несколько раз, но результат всегда был тот же, при загрузке выдавались сообщения:

```
Grub loading stage 1.5
Grub loading please wait
Error 18 <курсор>
```

Дальше дело не идет. Ко всему хорошему, в Windows я теперь тоже не могу загрузиться, и чтобы отослать это письмо, мне пришлось откопать старый компьютер.

Джеффри Бакл

О Поскольку под *Grub* отведено очень мало дискового пространства, сообщения об ошибках выдаются довольно невразумительные. «Error 18» означает «Выбранный цилиндр превышает



КРАТКАЯ СПРАВКА: PROC

Вы когда-нибудь просматривали файловую систему, разыскивая, кто сожрал всю память на диске, который вы когда-то считали бездонным? Если да, то вы, скорее всего, натыкались на директорию proc в корневом каталоге и спрашивали себя: а нельзя ли обойтись без нескольких тысяч файлов в **proc**? Многие на вид пусты, а в совокупности занимают, наверное, не меньше гигабайта.

Нет, удалять их нельзя. Порадуйтесь: места на жестком диске они не занимают, поскольку **/proc** – это виртуальная файловая система. На диске этих файлов вообще нет, это отображение состояния вашей системы, представленное в виде файлов для удобства работы с ним.

/proc содержит громадное количество псевдофайлов – на нашем компьютере их 5478, и в каждом ценнейшие данные. Выполните, например, команду **cat /proc/version**, и вы многое узнаете о вашем дистрибутиве и ядре. Назначение директорий **ide**, **scsi**, **bus/usb**, **bus/pci** и **bluetooth** очевидно, они содержат информацию об устройствах.

Вы можете также заметить директории с численными именами. Каждая такая директория описывает процесс, идентификатор (PID) которого и является названием директории. В каждой такой директории вы сможете найти такую информацию, как использование памяти, аргументы командной строки программы и т.п.

Большинство псевдофайлов в **/proc** предназначены только для чтения, однако в некоторые вы можете писать, на лету изменяя определенные параметры. Например, изменяя файл **/proc/sys/vm/swappiness**, вы можете контролировать интенсивность использования области подкачки (чем больше число, тем чаще происходит сброс страниц). Вы можете изменять этот файл следующим образом:

```
echo 30 >/proc/sys/vm/swappiness
```

если у вас есть исходные тексты ядра, вы найдете с информацию о **/proc**, открыв файл **/usr/src/linux/Documentation/filesystems/proc.txt**.

ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ DHPC

FAQ Что такое DHCP?
Это протокол динамической конфигурации узла (Dynamic Host Configuration Protocol).

FAQ Ну, это расшифровка, а нужен-то он зачем?
Когда вы подключаете компьютер к сети, вам нужно установить такие параметры, как IP-адрес, адрес шлюза и DNS-сервера. Чтобы не устанавливать их вручную (т.н. статическая адресация), а получать с сервера, и был придуман DHCP. Он упрощает работу в больших сетях и переезд из сети в сеть, как часто бывает с ноутбуками.

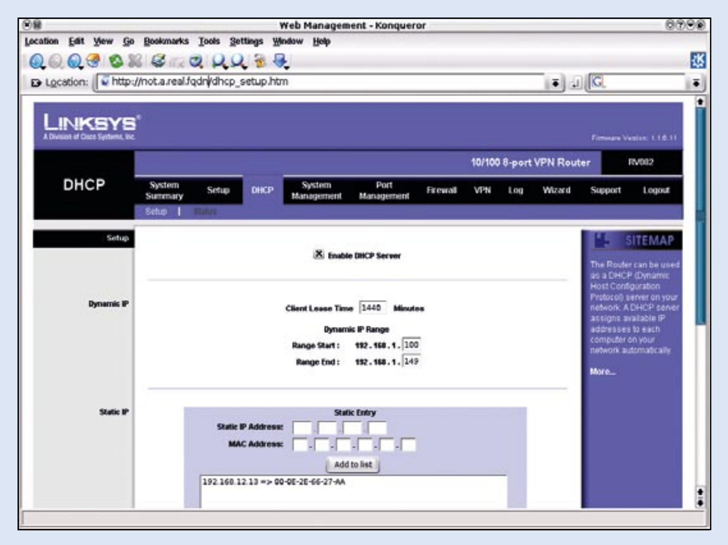
FAQ DNS? Что это такое?
Сервис доменных имен. Этот сервис нужен, чтобы переводить имя домена (**www.linuxformat.co.uk**) в IP-адрес (212.113.202.71).

FAQ Как он работает?
Клиент (ваш компьютер) посылает особый широковещательный «разведывательный» пакет. Этот пакет будет принят всеми компьютерами в сети, однако ответят на него только DHCP-серверы, прислав «предложение». Клиент выбирает, от какого сервера его принять, и запрашивает у него IP-адрес, шлюз и DNS.

FAQ А что за шлюз?
Шлюзом является компьютер, на который по умолчанию посылаются все пакеты, получатели которых расположены вне вашей локальной сети (например, в Интернете). Шлюз перенаправляет эти пакеты получателю и присылает ответ.

FAQ Где я могу взять DHCP-сервер?
Скорее всего он вам не понадобится. Большинство ADSL/Wifi-роутеров включают в себя DHCP-сервер. Кабельные модемы также используют DHCP. Если вы хотите запустить DHCP-сервер у себя в сети, ваш дистрибутив скорее всего включает в себя DHCP-сервер (клиент обычно устанавливается по умолчанию). Поищите DHCP в вашем пакетном менеджере. Или попробуйте Dnsmasq (**www.thekelleys.org.uk/dnsmasq**), разработку Саймона Келли (Simon Kelly), которым я сам пользуюсь.

FAQ Я где-то читал про DHCP-займ. Что это?
Когда вы разрываете сетевое соединение, ваш компьютер посылает сигнал DHCP-серверу об освобождении IP, и сервер может передать его кому-то другому. Если соединение прервалось некорректно (скажем, у вас внезапно погас свет или ваш ноутбук вышел из зоны беспроводного соединения), адрес будет по-прежнему занят. Этой проблемы можно избежать с помощью так называемого периода DHCP-займа, по истечению которого ваш клиент должен подтвердить, что адрес все еще используется. Если от клиента такого подтверждения не получено, адрес считается свободным.



Используем web-интерфейс роутера для настройки DHCP. Этот позволит вам присваивать машине адрес, необходимый серверу..

<< максимально поддерживаемый в BIOS». Другими словами, Ваш BIOS не справляется с жестким диском такой величины.

А Windows загружался потому, что его раздел находился в самом начале диска. То есть проблема кроется не в Linux, а в Вашем оборудовании. Все было бы так же, если бы Вы попытались установить вместе Windows XP и Windows 98.

Есть несколько вариантов решения этой проблемы. Можно уменьшить раздел с Windows (насколько, мы сказать не можем, потому что ничего не знаем о Вашем BIOS и жеском диске), и создать в SUSE отдельный раздел **/boot**. Благодаря этому файлы, требуемые *Grub*, попадут в начало раздела. Как только загрузчик заработает, все ограничения BIOS перестанут Вас волновать.

Однако лучше зайти на сайт изготовителя Вашей материнской платы и скачать самое свежее обновление BIOS.

Третий способ не идеален, но зато помогает сразу: можно загружаться с загрузочного CD SUSE, воспользовавшись опцией Boot from Hard Drive.

Чтобы восстановить загрузчик Windows, Вам надо загрузиться с установочного CD для Windows в режиме rescue и запустить

```
fdisk /mbr
```

(если у вас Windows 98) или

```
fixmbr
```

(для Windows XP/2000). ***НБ***

Ужимаем диск

Устанавливал SUSE Linux 10.0 с диска LXF 74. Процесс установки заглох на этапе разметки диска с ошибкой 3027 ('storage modification failed'). Это произошло, когда я попытался уменьшить размер раздела /dev/hda1 до 12.6 ГБ. У меня старый компьютер на базе Intel Celeron и Windows XP, которая установлена на 20-ГБ жестком диске, не имеющем разделов – и еще я абсолютный новичок во всех этих делах.

Кев Фрост

Думаю, написав, что на диске нет разделов, вы имели в виду, что на нем нет Linux-разделов. Ведь на нем должен быть хотя бы один Windows-раздел, иначе что установщик SUSE пытался уменьшить? Чаще всего подобная ошибка происходит потому, что раздел диска перед изменением размера не был дефрагментирован. В Windows перейдите в «Мой Компьютер», щелкните правой кнопкой на диск C:, выберите из контекстного меню «Свойства». Теперь перейдите на вкладку «Сервис» и проведите проверку диска, а затем дефрагментацию. После этого вставьте диск с SUSE и перезагрузите компьютер. Дефрагментатор Windows XP не особенно эффективен, поэтому Вам, возможно, придется запустить его несколько раз.

И еще, когда зайдете в Свойства диска в Windows, посмотрите, не установлена ли метка (имя) для этого раздела. Если метка есть, удалите ее. ***НБ***

Модем-импотент

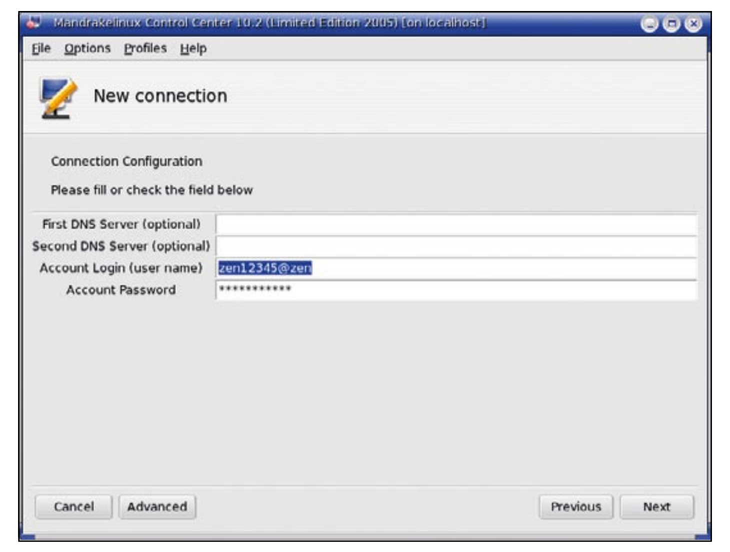
На своем компьютере я работаю в Windows XP и Mandriva Linux. На Windows Интернет от Tiscali (провайдер такой в UK,-прим. перев) работает хорошо. Но под Linux соединение не устанавливается, а в Tiscali не могут мне ответить. Не знаете ли вы, как в Linux настроить соединение через модем Sagem Fast 800? Индикатор питания горит, и даже иногда загорается индикатор сигнала, но сообщения об ошибках говорят о том, что модем не подключен.

На сайте Sagem посоветовали подключить модем к USB-хабу с дополнительным питанием, но по-моему, проблема не в питании – индикатор-то горит. Можете ли вы написать подробное пошаговое руководство для решения этой проблемы?

Джефф Биддулф

Лучшим решением будет замена USB-модема ADSL-модемом/роутером с Ethernet-интерфейсом, он Вам обойдется примерно в 35 евро. Обычно провайдеры предоставляют модем по минимуму – USB-модемы большую часть функций перекладывают на компьютер и нередко работают только с Windows. Аппаратный модем/роутер дает лучшую производительность и в Windows, и в Linux, а к тому же легко настраивается. Вдобавок Вы сможете подключить к нему несколько компьютеров.

Насчет дополнительного питания Вам сказали правильно. Спецификация USB требует тока всего в 500 mA (миллиампер) на один порт, что для современных USB ADSL-модемов в обрез. Индикатор питания еще ни о чем не говорит. При каждой установке соединения недостаток питания может вынудить модем зависнуть



Можно, конечно, установить драйверы Eagle для Sagem ADSL вручную, но *Центр Управления* Mandriva упростит эту задачу.

или разорвать соединение. У меня такого модема нет, но для ответа на Ваш вопрос я его позаимствовал и проверил его работу на ноутбуке с Mandriva. Работать-то он работал, но настолько медленнее моего Ethernet-модема, что я был в шоке.

Подключите модем и выждите пару секунд, пока загорятся индикаторы. Запустите *Центр Управления Mandriva*, перейдите в Сеть и Интернет, Создать новый сетевой интерфейс и выберите ADSL-соединение. В списке устройств должны быть USB-модемы Sagem. Быть может, Вам понадобятся установочные диски Mandriva для инсталляции драйвера, так что держите их под рукой. Теперь выберите провайдера из списка. В Великобритании всего два провайдера, но опция BT работает со всеми, кроме AOL, поскольку все используют BT-линии. Остальное – по умолчанию, кроме имени пользователя и пароля. Соединение может запускаться автоматически при каждом запуске системы, или вручную, с помощью скрипта *startadsl*. Проверьте имя пользователя и пароль, если соединение установить не удалось. ***НБ***

Запасная почта

Я располагаю выделенным сервером под Red Hat Enterprise Linux 4, где размещаются web-сайт и почта моей фирмы. Когда я только начал управлять сервером, у меня был только один домен, и стандартная конфигурация *Sendmail* работала хорошО потребовалось только добавить мой домен в /etc/mail/local-host-names и перезапустить *Sendmail*.

Мой бизнес быстро растет, и нагрузка на почтовый сервер становится все больше и больше – а если он рухнет, я начну терять прибыль. Дома у меня линия DSL и компьютер с Fedora Core 4. Подскажите мне, как использовать мой домашний компьютер в качестве резервного почтового сервера, если основной сервер откажет? ***Остин***

Допустим, у Вас домен **example.com**. В DNS нужно добавить две MX-записи:

```
example.com.        MX    10 primary.example.com.
example.com.        MX    20 secondary.example.com.
```

Так как у **primary.example.com** более низкий приоритет (**10**), все письма будут поступать на него. Зато если он вдруг «упадет», письма пойдут на **secondary.example.com**.

Изменения вступят в силу после перезапуска DNS-сервера. Убедитесь, что обе записи есть в домене.

На выделенном сервере ничего в настройке *Sendmail* изменять не придется. На Вашем домашнем компьютере потребуется создать файл **/etc/mail/relay-domains**, содержащий строку domain example.com. После создания файла перезапустите *Sendmail*. Убедитесь, что Ваш домен не содержится в файле **/etc/local-host-names** на **secondary.example.com**, иначе почта будет доставляться локально.

Я бы посоветовал Вам проверить работоспособность этой схемы: остановить Sendmail на выделенном сервере и отправить себе письмо с какого-нибудь стороннего сервера. Если настройки DNS корректны, письмо будет доставлено на Ваш домашний компьютер (запустите **tail** на **/var/log/maillog** для того, чтобы проследить за приемом почты). Не волнуйтесь, если письмо придет не сразу, ведь *Sendmail* сначала попытается достучаться до **primary.example.com**.

Sendmail на Вашем домашнем компьютере должен пересылать письма на

<< **primary.example.com** каждый час. Когда **primary.example.com** восстановится, письма с **secondary.example.com** через некоторое время на него прибудут. ***XX***

Нет контакта

В У меня проблемы с беспроводным соединением под Mandriva Linux 2006. Я использую карту D-Link Airplus G+ на ноутбуке (с *NdisWrapper*) для соединения с роутером D-Link G604T. Сначала все идет нормально, а в **/etc/resolv.conf** прописывается 212.30.8.150. Через 20-30 минут зайти на web-страницы становится невозможно. Соединение вроде бы не теряется, но когда я проверяю **/etc/resolv.conf**, оказывается, что в нем прописан 192.168.1.1, и мне приходится заново настраивать соединение.

Это происходит независимо от того, использую я WEP-шифрование или нет. У меня была подобная проблема в Mandriva Linux 2005, но там я всего лишь установил доступ только на чтение для файла **/etc/resolv.conf**, и проблема решилась. Однако с Mandriva Linux 2006 такой способ не сработал.

Дэйв Ле Хьюри, с форума LXF

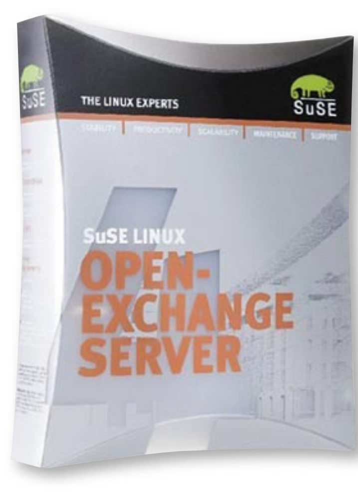
О Mandriva использует DHCP для получения сетевых настроек с роутера. Проблема в том, что на роутере запущен DHCP, а DNS-сервера там нет. Этот роутер, как и большинство, может работать в обеих ролях, однако DNS, по-видимому, отключен или неправильно настроен. Фактически, роутер сообщает компьютеру адрес DNS-сервера, который должен работать. В большинстве случаев достаточно перейти на вкладку DNS настроек роутера и настроить его на автоматический поиск. Если это не пройдет, укажите адреса DNS-серверов вручную.

Еще Вы можете запретить получение адреса DNS-сервера с DHCP. Для этого запустите *Центр Управления Mandriva*, следуйте в Сеть и Интернет > Настроить сетевой интерфейс. Выберите нужный интерфейс, перейдите на вкладку DHCP и выключите опцию автоматического получения DNS с DHCP. Установка прав **/etc/resolv.conf** только на чтение не спасет положение, потому что DHCP-клиент запускается с правами суперпользователя и может изменять любой файл в системе. ***НБ***

Коллективная работа

В У меня есть клиент, которому нужен корпоративный почтовый сервер, вроде *Microsoft Exchange*. Я, конечно, могу установить Exchange, однако мне известно, что можно настроить Linux в качестве контроллера домена. Мне нужно лишь знать, есть ли для под Linux такой почтовый сервер, работающий как *Exchange* с клиентом *Microsoft Outlook*? Было бы здорово, если нашлось решение целиком под Linux.

Тайво Акиношо



OpenExchange **унаследовал опыт и стабильность SUSE.**

О Есть несколько вариантов, все зависит от того, сколько денег готов вложить Ваш заказчик и какую работу Вы готовы проделать. *OpenGroupware.org* (**www.opengroupware.org**) – открытый groupware-сервер, работающий с любыми клиентами на всех распространенных платформах. Сам по себе он не является почтовым сервером, а предоставляет функции для работы со стандартными почтовыми серверами. *OpenGroupware.org* распространяется под лицензиями GPL и LGPL, а значит, платить за него Вам не придется. Однако будьте готовы к довольно сложной настройке.

Для серверов на отдельной машине *SUSE Linux Openexchange* обеспечивает эквивалент *MS Exchange* под Linux, и он работает с клиентами Microsoft как *Outlook*. Вы можете ознакомиться с подробной информацией об этом продукте на **www.novell.com/products/openexchange**. Учтите, что он небесплатен.

Еще одно решение – *Open-Xchange* (**www.openexchange.com**). Это коммерческий продукт, доступный для Red Hat и SUSE. Как и продукт SUSE, он полностью заменяет *MS Exchange*. Что предпочесть – зависит от бюджета Вашего заказчика. ***НБ***

Спасите наши серверы

В Один из наших Linux-серверов был недавно взломан. В итоге получился значительный простой, поскольку наш провайдер выключил сервер на время, пока наши специалисты все не исправят. Взлом не замечали около недели, пока третьи лица не прислали жалобу.

Почитав ваш журнал, мы пришли к выводу, что нужно следить за нашими серверами, и после взлома у нас на каждом сервере ежедневно запускается скрипт проверки. Можете ли вы посоветовать простой способ мониторинга открытых портов, который помогает обнаружить работу посторонних сервисов на наших серверах?

SC

О Безусловно, первой линией обороны должен быть аппаратный брандмауэр, либо *iptables*, настроенный на блокировку соединений по всем портам, кроме реально используемых.

Вместо запуска скрипта проверки на каждом сервере мы бы посоветовали использовать один из Ваших серверов для сканирования остальных при помощи какой-нибудь сторонней утилиты. Я предпочитаю *nmap* (**www.insecure.org/nmap**), потому что он умеет еще и проверять порты на прослушку. Простое сканирование сервера 192.168.100.100 по всем TCP портам в диапазоне 1-8000 будет запускаться так:

```
$ nmap -p 1-8000 192.168.100.100
```

Для облегчения обработки результатов можно попробовать скрипт вроде

```
nmap-audit (http://heavyk.org/nmap-audit).
```

В совокупности с *cron*, *nmap-audit* можно заставить посылать информацию о вскрытых портах на ваш электронный ящик. ***XX***

Белочкин ящик

В У меня дома работает почтовый сервер на SUSE с IMAP-сервисом *Cyrus* – лучше и быть не может. Письма приходят, проверяются двумя антивирусными сканерами и пересылаются на *Cyrus*. Таким образом я могу читать почту на домашнем компьютере и на ноутбуке, а с помощью *SquirellMail* – вообще на любом компьютере. Но... Недавно я случайно удалил полный почтовый ящик, и у меня нет резервной копии.

Как мне восстановить свой ящик и настройки? Я могу восстановить директорию /var/spool/imap, но это лишь вернет мои письма. Я могу сделать с помощью IMAP-клиента директорию внутри и скопировать туда почту. Однако это не вернет конфигурацию. Как ее восстановить?

Джероэн Роброк

О Восстановление конфигурации и восстановление писем – две отдельные задачи. Конфигурацию восстановить проще: она хранится в **/etc/imapd.conf** и **/etc/cyrus.conf**, так что если у вас есть резервные копии этих файлов, просто перепишите их.

Резервное копирование писем сводится к копированию директории **/var/spool/imap**, содержащей письма и информацию об их статусе. Я пользуюсь утилитой *rdiff-backup*, которая может сохранить копию директории как на локальной машине, так и на сетевой. Она также позволяет сделать откат на предыдущую версию, что прекрасно подходит для почтовых ящиков. Вы можете использовать *rdiff-backup* для регулярного резервирования вашей конфигурации. Поместите следующие строки в скрипт и вызовите его из *cron*:

```
$BACKUP_DIR=»/backup»
rdiff-backup --terminal-verbosity 2 /etc $BACKUP_DIR/etc
rdiff-backup --terminal-verbosity 2 /var/spool/imap $BACKUP_DIR/imap
```

Установите в качестве значения переменной **$BACKUP_DIR** каталог для резервных копий.

Если вы хотите хранить архивные копии ваших писем, можете попробовать *procmail* для копирования писем в резервный почтовый ящик. Что-то вроде

```
:0c:
$MAILDIR/${LOGNAME}-bak/
```

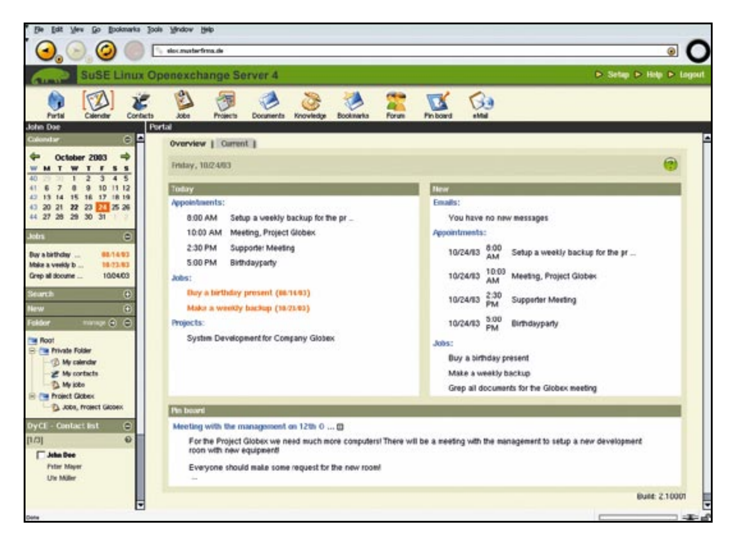
создаст резервную директорию для каждого пользователя и скопирует в нее все письма. ***НБ***

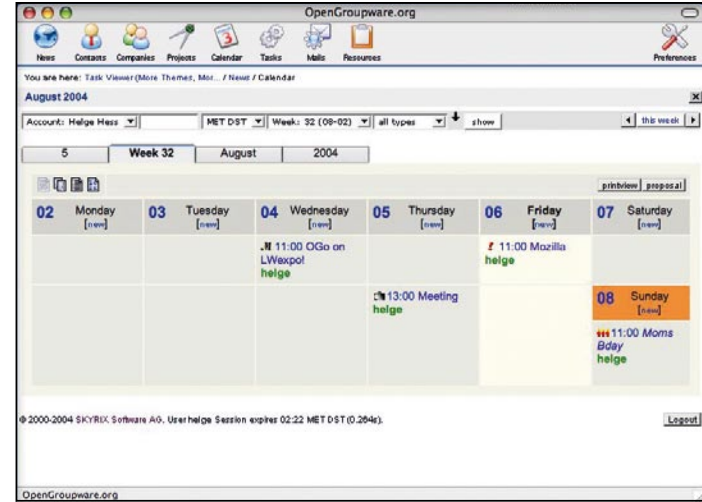
SUSE и ATi

В Здравствуйте. Вопрос такой. Мои приложения с графическим интерфейсом имеют одну раздражающую особенность: перерисовка их окон происходит слишком медленно. Проявляется это, в частности, так: при перетаскивании окна над другим окном за ним тянется «шлейф». Загрузка процессора при этом составляет порядка 80%.

Шлейф особенно заметен в приложениях, основанных на GTK (в Evolution меню открывается примерно







Novell SUSE поддерживает готовое groupware-решение (слева). ***OpenGroupware*** **(справа) бесплатен, но над настройкой придется поработать.**

секунду), в web-браузере Konqueror (именно когда он показывает web-станицу) и *Firefox*, а так же в *OpenOffice.org*. В приложениях, основанных на Qt, «шлейфа» нет, но загрузка процессора всё равно высока.

Загрузка процессора распределяется так: на долю *X*-сервера приходится порядка 55-65%, а остальное – на приложение, чьё окно перерисовывается. Аппаратная конфигурация у меня следующая: ноутбук Asus A2800S, P4-3GHz, 512 Mb RAM, ATI Mobility Radeon 9600. В Windows подобных проблем не наблюдалось. Я установил фирменные драйвера ATI с их сайта и включил поддержку *direct rendering*.

Операционная система SUSE 10, версия *Gnome – 2.12*. Версия KDE – 3.4.2, в качестве *X*-сервера используется *X.org* версии 6.8.2. Как можно решить такую проблему?

Алексей Титов

О Скорее всего, вы что-то упустили при установке драйверов. Это можно легко проверить: запустите glxgears в терминале, пусть она поработает 10-15 секунд. Если у вас меньше 1000 fps, то вероятно, X-сервер использует драйвер без поддержки аппаратного ускорения. На всякий случай я приведу полную последовательность действий, которые следует выполнить при установке драйверов Ati. Скачайте пакет с драйвером и перейдите в режим суперпользователя (командой su -). Проверьте, какие дистрибутивы поддерживаются скачанной вами версией драйвера:

```
sh ati-driver-installer-8.xx.xx-i386.run ?get-supported
```

Убедитесь, что в списке присутствует ваш дистрибутив (в данном случае вы должны отыскать строку SuSE/SUSE100-IA32). Если вы ее нашли, то можно создать RPM-пакет:

```
sh ati-driver-installer-8.xx.xx-i386.run ?buildpkg SuSE/SUSE100-IA32
```

Теперь остановите *X*-сервер и установите полученный пакет:

```
init 3
rpm -Uhv fglrx_6_8_0_SUSE-8.xx.xx.1.i386.rpm
```

Теперь нужно сообщить *X*-серверу, чтобы он использовал свежеустановленный драйвер. Можно отредактировать **/etc/X11/xorg.conf** вручную, однако в SUSE логичнее использовать *Sax*, тем более что в созданном нами пакете есть готовая Sax-конфигурация для драйвера ATi:

```
sax2 -r -m 0=fglrx -b /usr/share/doc/packages/fglrx/sax2-profile
```

Все готово, можно снова перейти в графический режим (**init 5**) и проверить, как работает аппаратное ускорение с помощью *glxgears* (теперь он должен выдавать никак не меньше 2000 FPS). ***AK***

Mandriva и TV-out

В Долгое время, примерно с появления *Mandrake linux 9.2*, я являюсь приверженцем именно этой линейки дистрибутивов.

С неделю назад появилась необходимость второго монитора, обратился к манам, к Интернету....

После выполнения ряда рекомендованных процедур, а именно:

1) скачивания с сайта **www.nvidia.com** рекомендованные драйвера.

2) установки модуля ядра согласно приложенному руководству.

3) правки файла **xorg.conf** (он находится во вложении в письмо, спорный момент там отмечен в комментарии к использованному драйверу, в нем приведены настройки для использования телевизора как второго монитора) на выходе получил ряд проблем.

1) невозможно поменять ориентацию второго экрана. *Option «TwinViewOrientation» «Clone» *при использовании других опций *X*-сервер вываливается с ошибкой.

2) при удачном запуске *X*-сервера и попытке выхода из него при стандартном «завершении сеанса» или при использовании Ctrl+Alt+BackSpace. получаем грустную картину, а именно попадаем не в текстовую консоль а в графический экран с разрешением 320 на 240 и размером шрифта (шрифтом это сложно назвать т.к. буквы сильно изуродованы) примерно в пол экрана....

Спорным моментом мне кажется является использование драйверов.

* Вариант1 – стандартный драйвер NV*

Результат не достигнут, но выход из X-сессии корректный.

* Вариант 2 – драйвер NVIDIA*

Неверная ориентация экрана, необходимо положение «Right», и некорректный выход из X-сессии.

Задача – использовать 2 монитора или монитор + телевизор, добиться корректного выхода из

Подскажите, пожалуйста, где искать ошибку?

Сергей Мариничев

О Использовать *TWinView* с драйвером nv, скорее всего, не получится. В лучшем случае будет доступен лишь режим клонирования, в худшем никакого изображения на втором мониторе/телевизоре вы не увидите. Если подключаете именно телевизор, то имеет смысл использовать лишь режим «Clone». Тогда обратите внимание на строку **Option «SecondMonitorVertRefresh» «60»**. У большинства телевизоров частота вертикальной развертки 50 герц (если, конечно, у вас не современный 100Гц-телевизор).

Если вы все-таки намерены использовать второй монитор (пусть для примера это будет такой же Samtron 56E), тогда внесите следующие поправки:

```
Option «TwinViewOrientation» «RightOf»
Option «ConnectedMonitor» «CRT,CRT»
Option «MetaModes» «832x624, 832x624»
```

Что касается некорректного завершения работы X-сервера, то тут скорее всего виновата строка Option «DPMS». Мы уже сталкивались с подобными проблемами: на некоторых видеокартах при включенном DPMS и нестандартных разрешений возникают подобные проблемы, так что вам придется либо закомментировать эту строку, либо сменить разрешение на стандартное (800x600). Ознакомьтесь с документом по адресу **http://linuxtech.pp.ru/modules/sections/index.php?op=viewarticle&artid=3**, чтобы поподробнее узнать о некоторых опциях драйвера nvidia.

Если аппаратное ускорение для вас непринципиально, и вы захотите использовать стандартный драйвер **nv**, то вместо *TwinView* следует обратить внимание на технологию *Xinerama* (эту надстройку можно применять с любой видеокартой, не только NVIDIA). Взять пример рабочего **xorg.conf** с настроенной *Xinerama* можно здесь: **http://www.opennet.ru/opennews/art.shtml?num=6921**. ***AK***

Неугомонный Linux

В Я установил SUSE 10 на свой ноутбук Asus A2800S. Я использую Gnome 2.12 в качестве графической среды (в KDE та же проблема). Дело в том, что когда я не произвожу никаких действий с компьютером (не трогаю его совсем), система всё равно постоянно обращается к диску. То есть индикатор работы диска на мгновение загорается раз в 3-5 секунд. Это не очень здорово, по скольку я хотел бы, чтобы диск «засыпал» после некоторого периода бездействия, особенно когда компьютер питается не от сети.

Я пробовал отключать **hald**, который был причиной такой проблемы в Fedora Core 4 (который был у меня был уста-



★ ВОПРОС-ПОБЕДИТЕЛЬ



Редакция вместе с компанией Bitrix выбрали самый интересный вопрос месяца и поздравляют победителя с заслуженной наградой – сертификатом на 1000 рублей в LinuxCenter.ru

ПРОСИМ ПОБЕДИТЕЛЯ ВЫЙТИ НА СВЯЗЬ С РЕДАКЦИЕЙ. НАШ АДРЕС INFO@LINUXFORMAT.RU

Тонкости лицензирования

Доброе время суток! Рассмотрим следующую ситуацию. Я пишу и продаю программы. Я был бы рад, если бы другие разработчики программ могли использовать части моего кода по своему усмотрению. Насколько я понимаю, я не могу использовать лицензию GPL, во-первых потому, что я сам продаю свои программы и, во-вторых, потому что я не хочу ограничивать других программистов рамками GPL.

Не хотел бы я ограничивать и приверженцев GPL. Я вообще с сомнением отношусь к идее авторского права (IP), в особенности по отношению к программам. Впрочем, законам я подчиняюсь. Я готов оказаться от авторства, но не хотел бы предстать перед судом из-за соответствия куска кода в моей и в чужой программе.

Известно, что большое количество уже существующих лицензий вносит путаницу и приводит к определённым трудностям на рынке ПО. Поэтому мне не хотелось бы изобретать нечто новое (да и нелегко это). Возможно, вы сможете подсказать, какая из лицензий соответствует поставленным условиям? Комментарий Эбена Моглена (Eben Moglen) к описанной ситуации был бы просто фантастикой!

Петр Васильев

Для начала, давайте проясним несколько моментов. Во-первых, GPL, как и любая другая лицензия, устанавливает условия использования вашего произведения (программы) другими людьми. При этом вас, как автора, не ограничивает ничего, кроме закона об авторском праве – вы можете продавать программу, не делиться частями кода и так далее. При этом существенно, чтобы все авторские права на программу принадлежали именно Вам. Если какой-нибудь человек захочет добавить в вашу программу хотя бы одну строчку кода, вам придется затребовать у него передачу авторских прав (по такой модели разрабатывается файловая система ReiserFS). Иными словами, до тех пор, пока вы владеете всеми правами на код программы, GPL и любая другая лицензия вас не ни в чем не ограничивает.

Во-вторых, GPL (и любая другая лицензия, признанная FSF свободной), не ограничивает продажу программы, то есть получение денег за физический акт передачи носителя с кодом другому лицу. Однако, при этом покупатель получает точно такой же набор прав, что и продавец – то есть он также может продавать эту программу или раздавать ее бесплатно, а также получает все исходные тексты и должен (в рамках GPL) делиться модификациями исходного кода.

В принципе, всем упомянутым в вашем вопросе требованиям удовлетворяет лицензия BSD, суть которой сводится к следующему: «Делайте с программой все, что вам заблагорассудится, но не удаляйте из исходных текстов уведомление об авторских правах». Лицензия BSD совместима как с GPL, так и с традиционными коммерческими соглашениями, кроме того, она закрепляет за вами факт авторства. Единственное, чего следует опасаться в данном случае – появления коммерческих продуктов, базирующихся на вашей программе, с продажи которых Вы не будете получать ни копейки. Если это для Вас неприемлемо, выпускайте свои исходные тексты по лицензии GPL.

В заключении отметим, что сотрудники редакции Linux Format не являются дипломированными юристами, поэтому, в случае возникновения сомнений вам следует обратиться к специалисту по авторскому и смежным правам. Не обязательно к Эбену Моглену – в мире много и других профессионалов самого высокого класса. ***ВС***



СПРАШИВАЙТЕ ПРАВИЛЬНО!

- Пожалуйста, не забывайте сообщать все необходимые данные о вашей системе. «У меня не работает X» нам мало что скажет, если мы не знаем, какую версию X вы имеете в виду и на какой аппаратной конфигурации пытаетесь ее запустить.
- Опишите свою проблему с максимальной точностью. Причитания типа «Оно не работает» или «У меня ошибка» вряд ли дадут нам возможность помочь Вам. Каким именно образом нечто не работает? Чего вы от него хотели? Какое поступает сообщение об ошибке?
- Помните, пожалуйста, что люди, работающие в нашем журнале, НЕ являются авторами или разработчиками Linux или Вашего конкретного пакета (дистрибутива). Иногда нужная Вам информация имеется на соответствующем сайте. Ознакомьтесь с документацией!

Мы стараемся ответить на все вопросы. Если вы не нашли ответа на свой, просмотрите другие ответы – может быть, там разбирается проблема, аналогичная вашей. Все вопросы, к сожалению, поместить невозможно.

новлен до этого), но это не помогло. Есть ли способ остановить это постоянное обращение?

Алексей Титов

Скорее всего, в «лишних» обращениях к диску виновата столь любимая наши британскими коллегами (и не только ими) программа-индексатор *Beagle*. Она отлавливает периоды простоя вашего компьютера и использует их, чтобы внести в базу данных новые документы. Таким образом, индексация файлов на жестком диске практически не влияет на «полезную» производительность системы. Некоторые коммерческие индексаторы следят за состоянием ноутбука и отключаются, если он переходит в режим питания от батарей. Не могу точно сказать, присутствует ли такая функциональность в *Beagle*, но, в случае необходимости, ее можно попытаться сэмулировать, написав простые сценарии-обработчики событий «battery.*» и «ac_adapter» демона **acpid**. В обработчике события **battery** следует остановить *Beagle*, а в сценарии **ac_adapte**r – вновь запустить его. Подробности ищите в документации к acpid. ***ВС***

LTSP

Доброго вам времени суток уважаемые работники мыши и клавиатуры! Есть локальная сеть с практически одинаковым оборудованием, имеется в наличии довольно серьезный компьютер (AMD Athlon 2600+, 512 RAM, 160Gb HDD), мониторы и видеокарты у всех одинаковые, вопрос: как мне сделать так чтобы компьютеры локальной сети грузились с главного компьютера и пользователи работали в графическом режиме? Желательно одновременно. ОС – ASP Linux 9.2 (планируется переход на ASP Linux 11). Второй вопрос – в софтверном супермаркете «Софткей» (**www.softkey.ru**) начался сбор предзаказов на ASP Linux 11, хотелось бы узнать что в ней нового будет поставлено и когда она выйдет... И последнее, в журнале частенько проходят сравнение программ, может вы сделаете обзор почтовых программ?

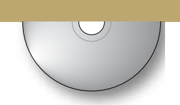
С уважением, Михаил.

Здравствуйте, Михаил. К сожалению, для подробного описания процесса установки терминал-сервера и настройке удаленной загрузке не хватит и двух журналов – настолько обширна эта тема. Почитать об этом можно на русскоязычном сайте *Linux Terminal Server Project* (**http://www.ltsp.ru**).

ASP Linux 11 уже вышел, и содержит в себе ядро 2.6.12, Xorg 6.8.2, KDE 3.5.0, Gnome 2.12. Express-версия дистри-







Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

FreeBSD 6.0
Серверная и настольная ОС
- Открытая
- Linux-совместимая
- Супернадежная

Включает: X.Org 6.8.2 * KDE 3.4.2
* Gnome 2.10.2 * Firefox 1.0.7
* OpenOffice.org 2.0

BSD BONANZA

OPENBSD 3.8
Непревзойденная защищенность

LINUX FORMAT 8 GIGABYTE DVD 5 FULL DISTROS





X.org 7.0 – новый модульный X-сервер
Астрономия – программы для любителей звездного неба
Xen 3.0 – виртуализация начинается здесь
Audacity 1.3 – звуковой редактор
Flock – новый «социальный» браузер
Игры - Snake 3D, Globulation, Sable



Содержание LINUX FORMAT DVD Март 2006

Сторона 1

Журнал
Answers
HotPicks
Roundup

Рабочий стол
AfterStep 2.2.0
Audacity 1.3.0
DosBox 0.63
PenguinTV1.0.4
X.Org 7.0

Разработка
Autopackage 1.0.8
Bazaar-NG 0.6.2
Boo 0.7.5.
KDevelop 3.3.1
Qt 4.1
WxAUI 0.92

Дистрибутивы
Open-Xchange LiveCD 1.0
Damn Small Linux 2.2b
DragonFly BSD 1.4.0
NetBSD 3.0
OpenBSD 3.8

Игры
Blockish 0.93
Globulation 0.8.17
HighMoon 1.2.3
Kye 0.5.0
NetHack 3.4.3
Sable 0.4
Snake3D 0.6
Vultures 1.11.2

Графика
Gimp 2.3.6
Inkscape 0.43
mtPaint 2.20

Справка
Linux Documentation Project.

Интернет
Azureus 2.4.0.0
BitTorrent 4.4.0
Flock 0.5.12
Guarddog 2.4.0
Sunbird 0.3a1

Офис
AbiWord 2.4.2
KBarcode 1.8.1
Morla 0.2
OpenOffice.org 2.0.2
Scribus 1.2.4

Сервер
Apache 2.2.0
IRCD-Hybrid 7.2.0
Icecast 2.3.1
MySQL 5.0.18
NMap 3.95
PHP 5.1.2
Samba 3.0.21a
Wildfire 2.4.0
XBNBT 81b.3
phpMyAdmin 2.7.0

Система
Conspy 1.3
GStreamer 0.10.1
Qemu 0.8.0
SUM 0.1.3
Xen 3.0


Поставляется вместе с журналом LINUXFORMAT номер 7(77) Март 2006

Сторона 2

FreeBSD 6.0 LinuxCenter Edition

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Кардмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8290. Лицензия МПТР России ВАФ № 77-68



СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.
iso
```

Чтобы упростить дальнейшее использование cdrecord, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку «Burn» и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на «Combust!». Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?
Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecrod* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?
А что если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не придется использовать Linux для записи дисков, подойдет любая операционная система, способная распознать привод записи дисков (см. выше).

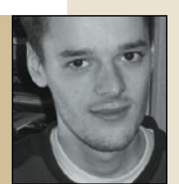
Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика.



Диск LinuxFormat

Хотите всесторонне изучить Unix? В этом месяце у нас на диске фестиваль BSD.

ЖМИТЕ НА СТАРТ



Майк Сондерс рад: BSD наконец-то в фокусе заслуженного внимания прессы!

За исключением некоторых игр, перенесенных из Windows, в данный момент под Linux довольно мало громких наименований – просто по причине малой доли, занимаемой Linux на рынке. Мы ждём у моря погоды, смиренно уповая, что крупные фирмы придут нас поразвлечь. А надо ли ждать?

«Брось, Майк», хмуритесь вы, «классные игры требуют больших групп разработчиков и дизайнеров». И вправду, хиты, сделанные одним-двумя разработчиками, встречаются редко. Но, как и серьёзные программные продукты, игры тоже могут быть в выигрыше благодаря вольной сети хакеров.

Бесспорно, лучше иметь команду программистов-энтузиастов, посвящающих все личное время любимому делу, а не персонал из нанятых кодировщиков, которые проводят каждый рабочий день в ожидании его конца.

Конечно, мы не скоро увидим очередную Zelda на SourceForge – все-таки крупномасштабные игры

> «Майк, – возразите вы, – классные игры требуют больших групп разработчиков.»

требуют серьёзных вложений денег и времени. Но программистам Open Source, не зависящим от прихотей рынка, никто не запретит опробовать свои радикальные идеи, хотя бы и нецелесообразные экономически. Отличные примеры таких игр – Tong и The Sheep Killer. Процесс разработки с открытым кодом – плодотворная почва для новых типов игр; посмотрим, куда заведёт мутация...

mike.saunders@futurenet.co.uk

СОДЕРЖИМОЕ DVD-ДИСКА



Сторона 1

Журнал

Answers Файлы из раздела «вопрос-ответов».
Article Index Список статей предыдущих номеров.
HotPicks Самые свежие приложения с открытым кодом.
Roundup Астрономическое ПО.

Рабочий стол

AfterStep Оконный менеджер.
Audacity Звуковой редактор.
DOSBox Эмулятор PC и DOS.
PenguinTV Читалка музыкальных и видеоблогов.
X.org Модульный X-сервер.

Разработка

Autopackage Среда для создания бинарных пакетов.
Bazaar-NG Система контроля версий.
Boo Язык программирования.
KDevelop Среда разработки.
Qt Инструментарий разработки.
WxAUI Библиотека GUI.

Дистрибутивы

DamnSmallLinux Легковесный дистрибутив.
DragonFlyBSD Ответвление от FreeBSD 4.x.
FreeBSD Высокопроизводительная операционная система.
NetBSD Кроссплатформенная система.
OpenBSD Безопасная операционная система.
Open-Xchange LiveCD Открытый аналог MS Exchange



Запустите свои старые игры под *DOSBox* – эмулятором PC и DOS

Игры

Blokish Настольная игра.
Globulation Стратегия реального времени.
HighMoon Бойня в стиле *Worms*.
Kye Игра-головоломка.
NetHack Исследование подземелий.
Sable Трёхмерная космическая стрелялка.
Snake3D Игра на быстроту реакции.
Vultures GUI для *NetHack*.

Graphics

Gimp Графический пакет (нестабильная версия).
Inkscape Векторный графический редактор.
MtPaint Программа рисования.

Офис

AbiWord Текстовый процессор.
KBarcode Дизайн и печать штрих-кодов.
Morla Редактор RDF-документов.
OpenOffice.org Офисный пакет.
Scribus Настольная издательская система.

Справка

LDP Проект документации Linux.

Интернет

Azureus Клиент *BitTorrent*.
BitTorrent P2P-обмен файлами.
BSFlite Минималистский клиент для AIM.
Flock Социальный' web-обозреватель.
Guarddog Утилита настройки брандмауэра.
Sunbird Календарь на базе *Mozilla*.

Сервер

Apache Web-сервер.
Icecast Потоковый аудиосервер.
IRCD-Hybrid IRC-сервер.
MySQL Сервер баз данных.
Nmap Сетевой сканер.
PHP Язык скриптов.
PhpMyAdmin Web-интерфейс к *MySQL*.
Samba Файловый и принт-сервер.
Wildfire Сервер обмена сообщениями *Jabber IM*.
XBNBT Трекер *BitTorrent*.

System

Conspy Удалённый мониторинг.
GStreamer Мультимедиа-обрамление.
Qemu Мультисистемный эмулятор.
Sum Менеджер обновлений системы.
Xen Монитор виртуальных машин.

Essentials

Allegro Библиотека для разработки игр.
Avifile Чтение и запись сжатых AVI-файлов.
CheckInstall Создание бинарных пакетов.
CSV Индексные файлы диска.
GLib Низкоуровневая сбиблиотека.
Glibc Библиотека GNU C.
GTK Инструментарий для создания GUI
Guile Встраиваемый язык Scheme.
Kernel Исходный код ядра и заплатки.
Libsigc Безопасные обратные вызовы для C++.
LibXML Анализатор XML для C и инструментарий.
Ncurses Инструментарий для создания текстовых интерфейсов.
Rawrite Запись образов на дискеты.
SDL Кросс-платформная мультимедиа библиотека.
SmartBootManager Системонезависимый загрузчик.
WxWidgets Библиотека GUI для C++.

Сторона 2

FreeBSD 6.0 LinuxCenter edition



***Sunbird* – кросс-платформенный календарь от Mozilla**

ДИСТРИБУТИВЫ

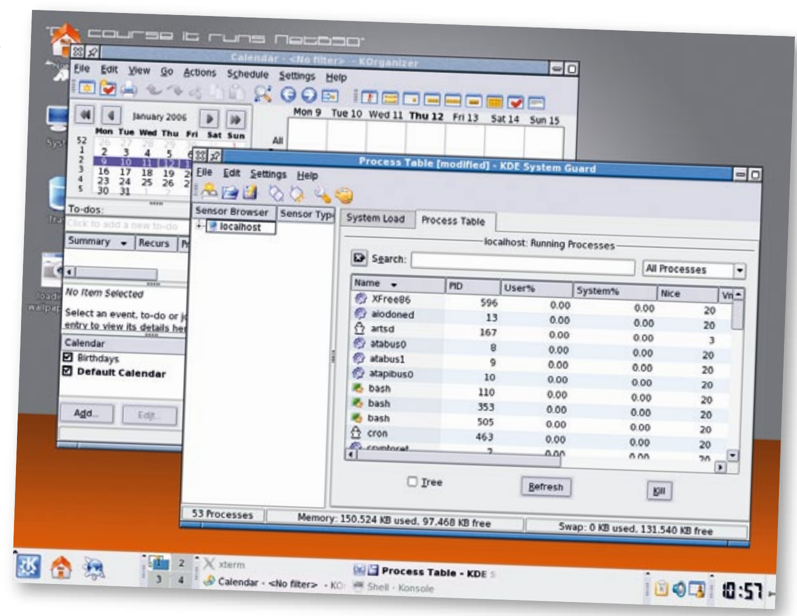
ЗОЛОТАЯ ЖИЛА BSD

Если вы уже наигрались с Linux и хотите вонзить зубы в другие Unix-системы, содержимое диска этого номера вам понравится. На DVD вы найдете FreeBSD и NetBSD, а также ISO-образы OpenBSD и нового дистрибутива DragonFly BSD. Все эти операционные системы выросли из BSD Unix – они зрелые и стабильные, и с их помощью можно создать отличную рабочую станцию или сервер. Однако их использование предполагает известный опыт работы в Unix-системах, так что если вы пока делаете первые шаги в Linux, лучше сначала проштудировать документацию. Тем, кому приходилось устанавливать Gentoo, Slackware или Debian, процедура установки не покажется слишком обременительной.

Мы не станем здесь описывать FreeBSD 6.0: эта система подробно рассмотрена на стр. 60. Достаточно сказать, что она надежна как скала и заслужила большое уважение на серверах x86 (под FreeBSD работает большинство машин Yahoo). А поскольку она основана на тех же стандартах Unix и POSIX, что и Linux, под ней отлично работает большинство программ Linux – процедура переноса очень проста. Из всех BSD систем, FreeBSD лучше всего подходит для PC, поскольку поддерживает широкий спектр программного обеспечения рабочих столов, а её инсталлятор проще аналогов из NetBSD или OpenBSD. DVD содержит оба диска FreeBSD.

Кроме того, мы включили на DVD ISO-образ NetBSD 3.0. Данная ОС – рекордсмен по числу поддерживаемых платформ: в текущем релизе их 57, и все из одного дерева исходных кодов! На нашем DVD вы найдете версии для x86, PowerPC Mac и Sun SPARC. NetBSD посложнее в установке, чем FreeBSD. В этой системе нет ни капли лишнего, а её опрятный дизайн превращает исследования и разработки с ней в удовольствие. Обратитесь к файлу **INSTALL.txt** (из каталога i386) за деталями установки и настройки NetBSD, а также к нашему обзору на стр. 28, чтобы побольше узнать о последнем релизе этой системы.

Стиль инсталлятора NetBSD – минимализм…



… но потерпите, и награда будет достойной!

ДИСТРИБУТИВЫ

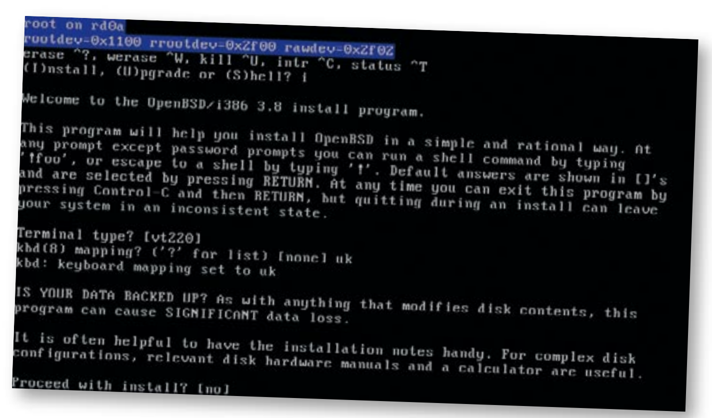
OPENBSD И DRAGONFLY BSD

НА ДИСКЕ

OpenBSD 3.8 на нашем DVD – это третья основная разновидность BSD-систем. Она отпочковалась от NetBSD в 1995-м году в результате разногласий, вынудивших Тео де Раадта (Theo de Raadt) начать свой собственный проект. С тех пор в ней делается акцент на безопасности. На web-сайте OpenBSD утверждается, что за восемь лет в системе была обнаружена только одна удалённая уязвимость. Такой уровень безопасности был достигнут в основном благодаря регулярному пересмотру кода и дотошному анализу добавляемых в систему возможностей. Хотя эту систему вполне можно использовать в качестве рабочей станции или сервера, OpenBSD чувствует себя дома на различных сетевых устройствах, типа маршрутизаторов и брандмауэров. Как и в случае NetBSD и FreeBSD, под ней можно запускать большинство Linux-программ, например, Apache или KDE. Инсталлятор OpenBSD полностью текстовый и управляется из командной строки, поэтому перед установкой важно прочесть файл **INSTALL.i386**.

Остается дистрибутив DragonFly BSD – один из самых свежих в линейке BSD. В 2003-м году Мэтт Диллон (Matt Dillon), разработчик FreeBSD и бывший кодер компилятора Amiga DICE, из-за трений с другими членами проекта FreeBSD решил создать ответвление от FreeBSD 4.x и направить его в другое русло, сфокусировавшись на работе над виртуальной файловой системой и симметричной многопроцессорностью. Его проект, DragonFly BSD, пока не рекомендуется для промышленного использования, но в нём уже есть достойный инсталлятор и вы можете сами оценить его прогресс, установив его с нашего DVD.

> «ЗА ВОСЕМЬ ЛЕТ РАБОТЫ У ОС ОБНАРУЖЕНА ВСЕГО ОДНА УЯЗВИМОСТЬ.»



Инсталлятор OpenBSD вообще спартанец: управляется командами.

ЭМУЛИРУЙ ЭТО!

Замечательно, что благодаря эмулятору PC *Qemu*, вы можете исследовать описанные системы без необходимости переразбивать свой жёсткий диск. Это отличный способ попробовать в деле BSD-систему для покамест неопытных пользователей, побаивающихся за свой жёсткий диск – он абсолютно безопасен. Во-первых, установите *Qemu* с нашего диска из раздела **System**, либо распаковав бинарный архив в свою корневую директорию /, либо скомпилировав исходные тексты командами **./configure**, **make** и **make install** (последняя – от имени root). После установки запустите команду

```
qemu-img create filesystem.img 1G
```

Создастся виртуальный жёсткий диск размером 1 ГБ, расположенный в файле **filesystem.img**. (Параметр **1G** можете заменить на '2G', или на '500M', или на какой угодно.) Затем запустите эмулятор *Qemu*, задав ему виртуальное устройство CD/DVD:

```
qemu -cdrom discimage.iso -hda filesystem.img -boot d
```

Qemu отобразит окно эмулируемого ПК, с файлом **discimage.iso** в качестве виртуального привода CD/DVD, а **filesystem.img** – в качестве жёсткого диска, и загрузкой с CD/DVD. (Если вы хотите загрузиться с настоящего CD/DVD-дисковода, воспользуйтесь командой **-cdrom /dev/cdrom**). После установки можно попросить *Qemu* загрузиться с виртуального жёсткого диска, командой

```
qemu filesystem.img
```

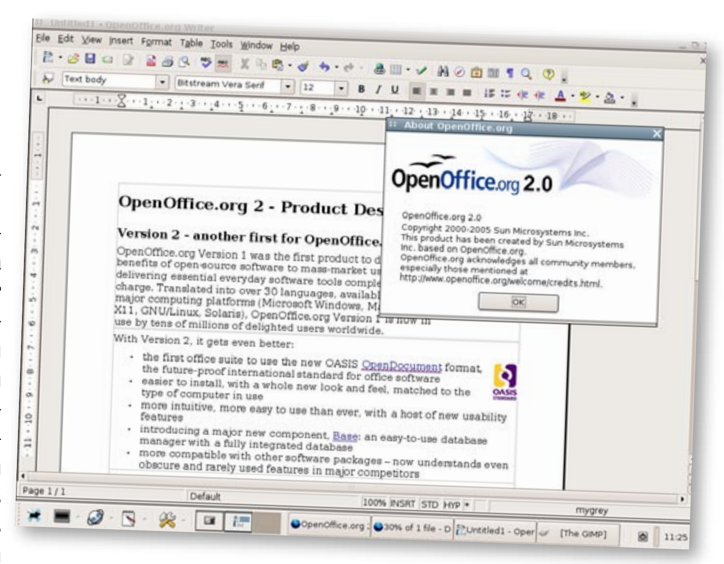
Проще некуда! Детали использования *Qemu* есть также на его man-странице.

НОВЫЕ РЕЛИЗЫ

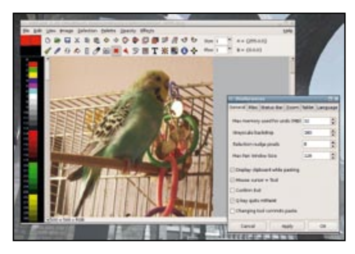
X.ORG 7.0 И OPENOFFICE.ORG 2.0.1

В декабре и январе наблюдался шквал новых релизов. *Apache* – самый популярный web-сервер в Интернете – подпрыгнул аж до версии 2.2 , о его новых возможностях читайте в нашем обзоре на стр. 24, а исходный код вы найдёте в разделе **Server** на DVD. На фронте рабочих столов произошёл долгожданный выпуск *X.org 7.0*. В этой версии исходный код наконец-то поделен на части. Новое модульное дерево *X* значительно облегчает процесс внесения улучшений в основу GUI в Linux. Код составлен из небольших секций, а значит, изменив что-либо в одно, не придется пересобирать всё. Мы включили весь исходный код и руководство по компиляции, но если процесс сборки *X* вам незнаком, возможно, для вашего дистрибутива будет лучше подождать официальных пакетов.

Вышла новая версия *OpenOffice.org 2.0.1*, исправившая груду недочетов и принесшая некоторое количество новых возможностей, например, возможность отключения части GUI. Разработчики *OOo* пока ещё не создали систему бинарных обновлений, так что вместо заплатки размером в 2 МБ пользователям придется скачивать все 100... Но мы записали релиз на наш диск, чтобы у вас не было проблем, и как обычно в случае официальных релизов *OOo*, они распространяются в виде архивов tar.gz с RPM-пакетами внутри. Пользователи Debian и Ubuntu могут сконвертировать их в пакеты Deb при помощи утилиты *Alien* (чтобы узнать, как пользоваться этой утилитой, наберите **man alien** в командной строке). Включен также *Xen 3.0*, обзор которого сделан на стр. 64.



Пока нет нормальной системы заплаток от *OpenOffice.org*, насладимся релизом 2.0.1, исправляющим ряд недочетов версии 2.0.



Если Gimp для вас перебор, попробуйте MtPaint, он проще.

ГРАФИКА

GIMP 2.3.6 И MTPAINT

Для художников в этом месяце мы припасли свежий срез *Gimp – 2.3.7*. Это – отличный способ увидеть в действии некоторые из новых возможностей будущего релиза серии 2.4. Улучшения коснулись времени загрузки, добавлены прокрутка и масштабирование в окнах пред-просмотра с помощью колеса мыши. Теперь вы можете копировать и вставлять SVG-изображения из *Gimp* в другие приложения, например, *Inkscape*. Добавлены также новые расширения, среди них – Recompose. Данный релиз не предназначен для повседневной работы, но по нашему мнению, его стабильности достаточно для освоения новых возможностей в ожидании версии 2.4. А если вы ищете чего-нибудь попроще, попробуйте облегченную альтернативу – *MtPaint*, приложение в духе *MS Paint*.

ДРУГИЕ ПРОГРАММЫ

Еще один дебютант этого месяца – *Qt 4.1*, обновление используемой в KDE библиотеки серии 4.x. Эта версия привнесла интегрированную поддержку векторной графики, систему печати на базе PDF и улучшения к пакету *Qt Designer*. Разработчики KDE особенно обрадуются появлению нового релиза *KDevelop* – см. обзор Грэма на странице 23.

По части настольных приложений, мы включили свежую версию *AfterStep*, не нового, но бравого оконного менеджера, сделанного под влиянием рабочего стола *NextStep* Стива Джобса (Steve Jobs). Его последние версии неплохо причесаны, и он по-прежнему намного легче, чем KDE и Gnome.

Если вы любите поболтать онлайн и ищете новый клиент AOL Instant Messenger, попробуйте *BSFLite*: он работает в текстовом режиме и идеален для запуска с терминала на медленных машинах или удалённых консолях, где нельзя запустить *X*. Одна из целей BSFLite – эффективность. Вы управляете клиентом с помощью команд – изучение потребует некоторого времени, но в результате позволит общаться гораздо быстрее и без перекидывания мыши по меню и панелям инструментов. Как сказал сам разработчик, «когда команды находятся на кончиках пальцев, уже не важно, помнит ли их ваш мозг».



Снимок экрана свидетельствует, что *NextStep*-подобный оконный менеджер *AfterStep* выглядит мило.

После 18 месяцев разработки вышел новый стабильный релиз *GStreamer* – мультимедиа-среды с открытым кодом, используемой в Gnome. Версия 0.10 принесла полную корзину новых возможностей и обновлений, включая улучшенную поддержку многопроцессорности и многопоточности, дробные частоты кадров для лучшей синхронизации аудио и видео, серьёзное улучшение производительности (кое-где почти вдвое быстрее, чем предыдущий релиз) и набор базовых классов, упрощающих жизнь разработчикам расширений. Релиз-заплатка 0.10.1, примчавшийся следом за 0.10, исправил некоторые мелкие проблемы и просчеты. Его исходный код находится в разделе **System** на DVD.

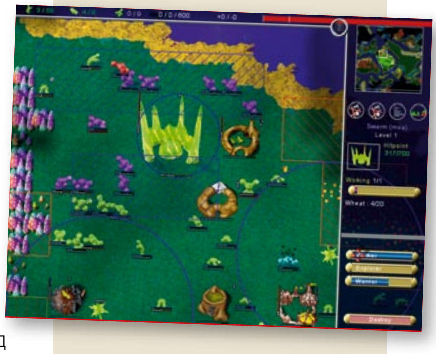
И НАКОНЕЦ ...

ИГРЫ

Восемь разных игр, на любой вкус, в составе диска этого месяца: стратегия, головоломка и аркады. Одна из изюминок – *Globulation*: это захватывающая стратегия реального времени, включающая собственный язык сценариев. Играть можно в локальной сети или против Интернет-игроков. *Globulation* требует библиотеку *Speex*, она есть в каталоге **Dependencies**. В зависимости от вашего дистрибутива вы можете установить её из пакетов RPM или Deb или собрать из исходных текстов.

Если хотите чего-то поактивнее, попробуйте *Sable* – стилизованную 3D-стрелялку, созданную под впечатлением от *After Burner* и *Space Harrier*. Смотреть там особо нечего, но довольно забавно пострелять, к тому же поражает, как много разработчик сумел уместить в небольшом архиве размером 211K. Фанаты традиционных игр Unix насладятся игрой *Vultures* – приятный графический интерфейс для ветерана *NetHack*. (*NetHack* – игра с походами и сражениями в подземельях, с симпатично-простецким текстовым представлением героев и злодеев).

Захотелось напрячь мозги – взгляните на *Kye*. В этой головоломке нужно решать, как расположить объекты, раскиданные по экрану. И, наконец, игра с замечательным именем *Blokish*. Она не имеет никакого отношения к распитию пивка и обсуждению вчерашнего матча. Это клон настольной игры *Blokus* для четырёх игроков. Ваша задача – избавиться от двадцати одной фишки, не дав остальным игрокам сделать то же самое. В игру также входит противостояние базовому компьютерному игроку. LXF



Гротескные изображения в военной стратегии *Globulation*.

Red Hat Enterprise Linux 4 - надежная платформа для вашего бизнеса!

Red Hat Enterprise Linux 4 открывает новую эпоху в истории корпоративных систем от компании Red Hat и значительно превосходит по возможностям своего предшественника - Red Hat Enterprise Linux 3. Продукт обеспечивает повышенную безопасность, производительность и масштабируемость, но при этом сохраняет совместимость с более ранними выпусками. Использование единой кодовой базы для всех редакций Red Hat Enterprise Linux означает, что один и тот же комплект дистрибутивных дисков можно использовать для установки на платформах x86, Intel EMT64 и AMD64.

Red Hat Enterprise Linux 4 Advanced Server



Red Hat Enterprise Linux 4 Advanced Server (AS) - прекрасная платформа для поддержки баз данных, корпоративных приложений ERP/CRM и решения бизнес-критичных задач. Это единственная редакция Red Hat Enterprise Linux, способная работать на высокопроизводительных серверах серии IBM Power и мейнфреймах IBM S/390 или zSeries. Red Hat Enterprise Linux 4 AS не имеет ограничений на объем оперативной памяти и число процессоров, помимо накладываемых аппаратной частью, и сертифицирован ведущими поставщиками ПО, например, корпорацией Oracle.

Уровень обслуживания "Standard". Поддержка через Web или по телефону в режиме 12x5. Время реакции, не превышающее 4 бизнес-часа. Доступ к сети Red Hat Network в течение одного года.
Уровень обслуживания "Premium". Поддержка через Web или по телефону в режиме 24x7. Время реакции, не превышающее один бизнес-час. Доступ к сети Red Hat Network в течение одного года.

Red Hat Enterprise Linux 4 Entry-level Server



Red Hat Enterprise Linux 4 ES ориентирован на использование в качестве сервера отдельного департамента или небольшой организации. С его помощью можно легко развернуть web-сервер, службу печати или файловый сервер. Red Hat Enterprise Linux 4 ES может работать на системах, имеющих до двух процессоров и до 16Гб основной памяти.

Уровень обслуживания "Standard". Поддержка через Web или по телефону в режиме 12x5. Время реакции, не превышающее 4 бизнес-часа. Доступ к сети Red Hat Network в течение одного года.
Редакция Basic Edition обеспечивается 30-ти дневной технической поддержкой по вопросам инсталляции и настройки системы (по телефону или через Web) и годичной подпиской на услуги сети Red Hat Network.

Red Hat Enterprise Linux Workstation



Red Hat Enterprise Linux WS - многофункциональный клиент для разработки программного обеспечения, CAD-приложений и решения других технических задач. Система может работать как на одно, так и на двухпроцессорных рабочих станциях, что позволяет в полной мере задействовать возможности имеющегося оборудования.

Уровень обслуживания "Standard". Поддержка через Web или по телефону в режиме 12x5. Время реакции, не превышающее 4 бизнес-часа. Доступ к сети Red Hat Network в течение одного года.
Редакция Basic Edition обеспечивается 30-ти дневной технической поддержкой по вопросам инсталляции и настройки системы (по телефону или через Web) и годичной подпиской на услуги сети Red Hat Network.

Заказывайте Red Hat Linux на сайте Линуксцентра : www.linuxcenter.ru



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно
Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:

ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru

Переводчики
Юрий Афанасьев, Александр Бикмеев, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Салимов, Сергей Супрунов, Александр Толстой, Александр Черных, Юлия Шабунио, Павел Шер.
Литературные редакторы
Родион Водейко, Иван Мищенко, Елена Толстякова
Допечатная подготовка
Мария Пучкова
Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Винниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Павел Фролов, Станислав Медведев
Отпечатано в типографии "Текст",
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейтч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор
Джулиан Джефферсон (Julian Jefferson) julian.jefferson@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Операционный редактор
Ребекка Смелли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрью Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

АВТОРЫ
Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Энди Чэннел (Andy Channelle), Дэвид Картрайт (David Cartwright), Ричард Коббетт (Richard Cobbett), Алекс Кокс (Alex Cox), Марко Фиоретти (Marco Fioretti), Джо Харрис (Jo Harris), Майкл Микс (Michael Meeks), Дмитрий Кирсанов, Александр Бикмеев, Александр Кузьменков, Владимир Попов, Сергей Супрунов, Сергей Яремчук

ХУДОЖЕСТВЕННЫЙ ОТДЕЛ
Арт-ассистенты: Джон Блекшау (John Blackshaw), Симон Годдард (Simon Goddard), Дэниэль Винсент (Daniel Vincent)
Фотографии: Joby Sessions, Simon Lees, Superstock, Photodisc, Photolibrary.com
Иллюстрации: Пол Бейтмен (Paul Bateman), Крис Винн (Chris Winn), Шейн Коллин (Shane Collinge)
Создание диска: Майк Сондерс (Mike Saunders)

КОНТАКТНАЯ ИНФОРМАЦИЯ
England: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email linuxformat@futurenet.co.uk
Россия: Санкт-Петербург, ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Email: info@linuxformat.ru
Web: www.linuxformat.ru

В следующем месяце

LINUX FORMAT 4 (78) АПРЕЛЬ 2006

ЦИФРОВОЙ РАЙ

Учимся снимать, скачивать, редактировать и печатать шедевры – только с Open Source

The exact contents of future issues are subject to change

LXF-ИНТЕРВЬЮ

Люди в Mono

Встречайте: Эдд Дамбилл (Edd Dumbill) и Нил Борнштейн (Niel Bornstein)

Web 2.0 – мифы и реальность

Как менталитет Open Source может изменить мир

Syllable OS

50 Мб, загружающиеся за 10 секунд!